Digitale Wasserzeichen zum Integritätsschutz von Videodaten

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)

vorgelegt von

Dipl.-Inform. (FH) Stefan Thiemert

geboren in Dessau

Referenten:	Prof. Dr. Stefan Katzenbeisser, Darmstadt
	Prof. Dr. Jana Dittmann, Magdeburg
	Prof. Dr. Rüdiger Grimm, Koblenz
Tag der Einreichung:	13. März 2013
Tag der mündlichen Prüfung:	29. April 2013

Darmstadt 2013

Hochschulkennziffer D17

Wissenschaftlicher Werdegang des Verfassers[1]

10/1997–03/2002 Studium der Informatik an der Hochschule Anhalt (FH), Köthen

04/2001–03/2002 Diplomarbeit am Fraunhofer IPSI, Darmstadt
„Werkzeuge zur Qualitätsevaluierung und Vorschläge zur Optimierung von MPEG-Videowasserzeichen“

04/2002–03/2004 Wissenschaftlich-technischer Mitarbeiter am Fraunhofer IPSI, Darmstadt

04/2004–12/2006 Wissenschaftlicher Mitarbeiter am Fraunhofer IPSI, Darmstadt

01/2007–01/2008 Wissenschaftlicher Mitarbeiter am Fraunhofer IGD, Darmstadt

02/2008–09/2010 Wissenschaftlicher Mitarbeiter am Fraunhofer SIT, Darmstadt

05/2011–09/2011 Wissenschaftlicher Mitarbeiter am Fraunhofer SIT, Darmstadt

seit 10/2011 Studium der evangelischen Theologie an der Theologischen Hochschule, Dietzhölztal-Ewersbach

[1] Gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt

Abstract

In this work the need for the protection of the integrity of videos will be discussed. Several scenarios will be defined where the integrity of the data is necessary, e.g. in the case of surveillance. A concept is being designed, which embeds with a robust watermarking scheme content-describing features.

As base for the concept an existing robust watermarking scheme is being improved. The scheme has a capacity of 64 Bits/s and a good robustness against lossy compression in combination with format conversion and scaling.

In this work we will develop four methods for creating feature vectors. Two of these methods project non-content dependencies to feature vectors while the other methods are based on content-dependent features delivered by interest operators. We will show that content- as well as non-content dependencies are suitable for the usage.

We will test this methods concerning their robustness against content-preserving and their sensitivity against content-changing methods. The test results confirm the good robustness of the basic robust watermark. In the same way the good sensitivity of the content-decribing feature is being confirmed.

Zusammenfassung

Im Rahmen dieser Arbeit wird die Notwendigkeit des Integritätsschutzes von Videodaten diskutiert. Es werden verschiedene Szenarien definiert, in denen die Unversehrtheit der Daten benötigt wird, z.B. im Rahmen von Videoüberwachung. Es wird ein Konzept entworfen, welches mit Hilfe eines robusten Wasserzeichenverfahrens inhalts-beschreibende Merkmale in das Video einbettet.

Als Basis für das Konzept wird ein bestehendes robustes Wasserzeichenverfahren weiterentwickelt. Das Verfahren verfügt über eine Kapazität von 64 Bit/s und zeigt gute Robustheit gegenüber verlustbehafteter Kompression mit Formatumwandlung und Skalierung.

Im Rahmen dieser Arbeit werden vier Verfahren zur Generierung von Merkmalsvektoren entwickelt. Zwei dieser Verfahren bilden nicht-inhaltliche Abhängigkeiten auf Merkmalsvektoren ab, während die beiden anderen Verfahren auf inhaltlichen Abhängigkeiten durch Interest-Operatoren basieren. Wir zeigen, dass sowohl inhaltliche als auch nicht-inhaltliche Abhängigkeiten zur Verwendung geeignet sind.

Die Verfahren werden verschiedenen Tests hinsichtlich ihrer Robustheit gegenüber inhalts-erhaltenden und ihrer Sensitivität hinsichtlich inhalts-verändernden Maßnahmen unterzogen. Die Testergebnisse bestätigten die gute Robustheit des zugrundeliegenden Wasserzeichenverfahrens. Ebenso wird eine gute Sensitivität des Verfahrens bestätigt.

Danksagung

Eine Dissertation ist kein Ein-Mann-Projekt. Ohne die Unterstützung vieler Leute in all den Jahren wäre dieses Projekt sicher nicht zu einem erfolgreichen Abschluss gelangt. Aus diesem Grund möchte ich hier einige Personen erwähnen, denen ich sehr dankbar bin.

Prof. Dr. Stefan Katzenbeisser danke ich für die Betreuung dieser Dissertation. Auch Prof. Dr. Jana Dittmann und Prof. Dr. Rüdiger Grimm haben viel Zeit in die Korrektur des Manuskriptes und in die Beratung investiert.

Der Mediensicherheits-Arbeitsgruppe des Fraunhofer SIT um Dr. Martin Steinebach danke ich für die jahrelange Unterstützung. Sie haben mich gerade durch die anregenden Diskussionen unterstützt und inspiriert. Darüber hinaus haben mich während meiner Tätigkeit viele Studenten im Rahmen von Diplom-, Bachelor-, Masterarbeiten und Praktika unterstützt.

Ich danke meiner Frau Rahel für ihre Rücksichtnahme, für den Freiraum, den sie mir gegeben hat und für ihre Motivation. Darüber hinaus danke ich meiner Familie für ihre Unterstützung.

Ein großer Dank gilt meinen Freunden, hier insbesondere Christian Spiertz und Christopher Janke, meinen Kommilitonen und Dozenten an der Theologischen Hochschule in Dietzhölztal-Ewersbach. Danke für die praktische Unterstützung gerade in der Endphase dieser Dissertation und für die Unterstützung im Gebet.

Zum Schluss aber vor allem danke ich meinem Gott, durch den ich bin, wer ich bin und durch den all dies möglich wurde. Ihm allein sei die Ehre.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Problemstellung und Begriffsklärungen

1.1 Problemstellung

Die Menge digitaler Videos wächst ständig. Täglich werden Massen von Videos generiert und über das Internet übertragen. Darüber hinaus existiert mittlerweile eine Vielzahl auch frei verfügbarer Programme mit denen jeder Videos bearbeiten kann. Dadurch ergibt sich eine neue wissenschaftliche Herausforderung: der Nachweis der Unversehrtheit des Inhalts.

Zur Überprüfung des Videoinhalts wurden in der Vergangenheit verschiedene Methoden entwickelt: digitale Signaturen und kryptographische Hashfunktionen, digitale Wasserzeichen und forensische Methoden.

Ziel dieser Arbeit ist es Ansätze zum Schutz des Inhalts digitaler Videos zu entwickeln, die auf der Basis digitaler Wasserzeichen arbeiten. Das zu entwickelnde Verfahren soll einen Merkmalsvektor aus dem Inhalt eines Videos generieren und diesen mittels eines robusten Wasserzeichenverfahrens in das zu schützende Video einbetten. Veränderungen am Inhalt sollen angezeigt werden bei gleichzeitig angestrebter Formatunabhängigkeit. In Kapitel 3 zeigen wir, dass bestehende Verfahren überwiegend entweder formatabhängig sind oder das Video komplett authentifizieren, d.h. Veränderungen nicht lokalisieren sondern komplette Frames als verändert identifizieren.

Zunächst definieren wir den Begriff „Inhalt eines digitalen Videos“, der in dieser Arbeit verwendet wird:

Unter dem Inhalt eines digitalen Videos verstehen wir die Kernaussage der sichtbaren Informationen. Ändert eine Operation am Videomaterial die Kernaussage der sichtbaren Informationen so definieren wir sie als inhaltverändernde Operation.

Beispiele für inhaltsverändernde Operationen werden in Kapitel 2 behandelt. Welche sichtbaren Informationen relevant für die Kernaussage sind ist abhängig vom jeweiligen Kontext und Anwendungsszenario.

1.2 Digitale Wasserzeichen

Nach Dittmann [Dit00] verstehen wir unter einem digitalen Wasserzeichen ein nichtwahrnehmbares Signal, das durch einen Einbettungsalgorithmus E in ein Dokument C eingebracht wird. Der Einbettungsalgorithmus wird mit der einzubettenden Information I und dem geheimen Schlüssel K gesteuert und liefert als Ergebnis das markierte Dokument $C_W = E(C, I, K)$ zurück. Im Ausleseprozess wird die eingebettete Information $I = R(C_W, K)$ mit Hilfe des Auslesealgorithmus R aus dem markierten Dokument C_W ausgelesen.

Dittmann definiert in [Dit00] Verfahrensparameter, nach denen Wasserzeichenverfahren eingeordnet werden können:

- *Robustheit* beschreibt die Auslesbarkeit des Wasserzeichens nach Veränderungen am Material, wie z.B. Re-Enkodierung oder Formatumwandlung.
- *Nicht-Detektierbarkeit* bedeutet, dass ein Angreifer auch bei Kenntnis des Originals nicht auf ein Wasserzeichen schließen kann.
- *Nicht-Wahrnehmbarkeit* bedeutet, dass das Wasserzeichen für das menschliche Wahrnehmungssystem nicht erkennbar sein darf.
- *Sicherheit* beschreibt die Auslesbarkeit des Wasserzeichens nach absichtlichen Veränderungen am Material mit Kenntnis des Wasserzeichenverfahrens.
- *Komplexität* ist der Aufwand zum Einbetten und Auslesen eines Wasserzeichens.
- *Kapazität* beschreibt die Menge an Informationen, die durch das Wasserzeichenverfahren eingebettet werden kann.
- *Verifikationsmethode* unterscheidet zwischen öffentlicher (öffentlicher und privater Schlüssel) und geheimer Verifikation (nur privater Schlüssel).
- *Invertierbarkeit* beschreibt die Möglichkeit das Wasserzeichen bis zu einem gewissen Grad aus dem markierten Dokument zu entfernen.

Cox et al. definieren in [CMB+07] das in Abbildung 1.1 dargestellte Klassifizierungsschema für Integritätswasserzeichen. Wir definieren in dieser Arbeit die unterschiedlichen Typen der Integritätswasserzeichen wie folgt:

- Exakte Authentifizierung

  - *Fragile Wasserzeichen* erkennen sämtliche Manipulationen am Datenmaterial.
  - *Invertierbare Wasserzeichen* erkennen ebenfalls alle Manipulationen, bieten jedoch die Möglichkeit der Wiederherstellung des ursprünglichen Datenmaterials.

- Selektive Authentifizierung
  - *Semifragile Wasserzeichen* können zwischen erlaubten und nicht erlaubten Manipulationen unterscheiden. Die Definition von erlaubten und unerlaubten Manipulationen ist dabei vom Anwendungsszenario abhängig.
  - *Inhaltsfragile Wasserzeichen* können ebenfalls zwischen erlaubten und nicht erlaubten Manipulationen unterscheiden. Sie bilden eine Spezialisierung der semifragilen Wasserzeichen, da sie Informationen über den Inhalt als Einbettungsmerkmal verwenden.

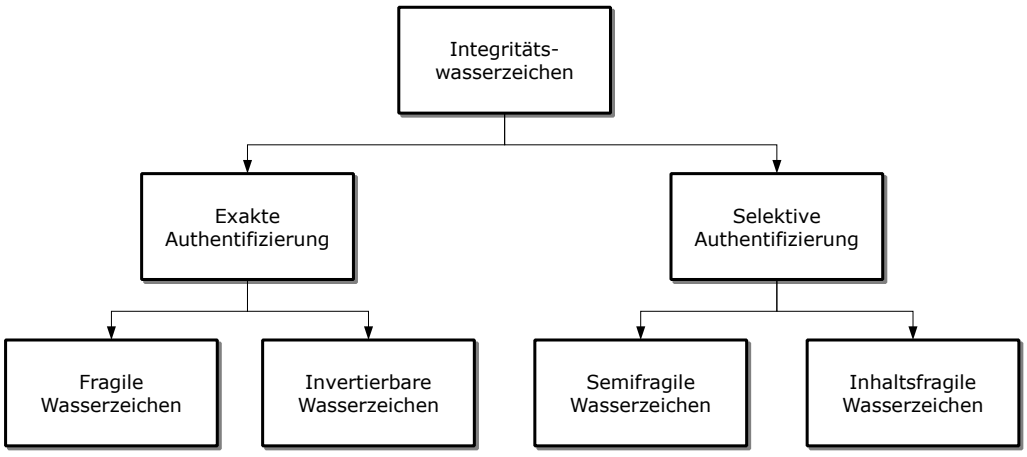


Abbildung 1.1: Klassifizierungsschema von Integritätswasserzeichen

1.3 Verwandte Disziplinen

Erste Ansätze zum Nachweis der Unversehrtheit digitaler Daten finden sich in der Kryptographie [Sch96]. Kryptographische Hashfunktionen bilden Zeichenketten beliebiger Länge x auf Zeichenketten fester Länge $h(x)$ ab [Buc10]. Sie haben die Eigenschaft, dass bereits eine minimale Veränderung in x den Hashwert $h(x)$ komplett verändert. Eine kryptographische Hashfunktion sollte folgende Eigenschaften besitzen:

- *Kollisionsresistenz*: Es sollte praktisch unmöglich sein zwei Eingabezeichenketten x und y zu finden, für welche gilt: $h(x) = h(y)$ mit $x \neq y$.

- *Einwegfunktion*: Es sollte unmöglich sein zu einem Hashwert $H(x)$ eine zugehörige Eingabezeichenkette x zu ermitteln.

Viele Verfahren zur Erstellung digitaler Signaturen arbeiten nach dem folgenden Prinzip: Ihnen dienen Hashfunktionen als Basis. Zum Signieren eines digitalen Dokumentes wird der Hashwert des Dokumentes mit dem privaten Schlüssel des Senders verschlüsselt. Um die Echtheit des Dokumentes auf Empfängerseite zu prüfen wird der übermittelte Hashwert mit dem öffentlichen Schlüssel des Senders entschlüsselt und mit dem Hashwert des übermittelten Dokuments verglichen. Dadurch kann sowohl die Identität des Senders als auch die Unversehrtheit des Dokuments überprüft werden [Sch96]. Da Hashfunktionen bereits auf minimale Änderungen des Inhalts mit großen Änderungen des Hashwertes reagieren ist eine digitale Sigantur ebenfalls nach minimalen Änderungen ungültig und in diesem Fall das Video nicht authentisch.

Neben kryptographischen Hashfunktionen besteht die Möglichkeit den Inhalt von Dokumenten beliebiger Länge mit Hilfe robuster Hashfunktionen auf Hashwerte fester Länge abzubilden. Statt der binären Repräsentation des Dokuments bilden robuste Hashfunktionen den Inhalt auf Hashwerte ab. Robuste Hashfunktionen wurden ursprünglich für das schnelle Auffinden von ähnlichen Daten in großen Datenbeständen entwickelt. Im Gegensatz zu kryptographischen Hashfunktionen verfügen robuste Hashfunktionen nicht über eine starke Kollisionsresistenz. Kleine Veränderungen am Dokument führen nicht zu großen Veränderungen am Hashwert. Ein robustes Hashverfahren für Videos stellen Haitsma et al. in [OKH01] vor.

Eine dritte Disziplin zum Erkennen von Manipulationen an digitalen Daten ist die digitale Forensik. Sie verfolgt den Ansatz Veränderungen am Datenmaterial zu erkennen ohne Informationen über das ursprüngliche Material zu verfügen. Forensische Verfahren für Bild- und Videodaten werden u.a. von Hany Farid erforscht [PF04][WF06].

1.4 Aufbau der Arbeit

Die Arbeit gliedert sich in sechs Hauptteile. In Kapitel 2 definieren wir verschiedene Anwendungsszenarien für digitale Videowasserzeichen zum Integritätschutz. Dabei werden drei Szenarien vorgestellt und ihre jeweiligen Anforderungen definiert. In Kapitel 3 gehen wir auf verschiedene Lösungsansätze ein, welche die Unversehrtheit von Bildern und Videos erkennen können. Dabei gehen wir auf Vor- und Nachteile dieser Techniken ein. In Kapitel 4 stellen wir unseren Entwurf vor und stellen anschließend in den Kapiteln 5 und 6 von uns entwickelte Ansätze von robusten Videowasserzeichen und inhalts-beschreibenden Merkmalen vor. In Kapitel 7 evaluieren wir schließlich unser Konzept aus Kapitel 4.

Das von uns entwickelte robuste Videowasserzeichen verfügt über eine gute Robustheit gegenüber inhalts-erhaltenden und inhalts-verändernden Maßnahmen. Dabei

hat es eine hohe Kapazität (64 Bit/s). Das inhalts-beschreibende Merkmal kann zwischen inhalts-erhaltenden und inhalts-verändernden Maßnahmen unterscheiden und letztere lokalisieren.

Kapitel 2

Anwendungsszenarien

Dieses Kapitel befasst sich mit verschiedenen Anwendungsszenarien bei denen Integritätswasserzeichen zum Einsatz kommen können. Dazu beschreiben wir zunächst Beispiele für die Anwendungsfälle und definieren daraus Anforderungen für die Integritätswasserzeichen.

2.1 Videoüberwachung

Videoüberwachungssysteme werden im öffentlichen und nicht-öffentlichen Sektor eingesetzt. Zum öffentlichen Sektor zählen wir u.a. Straßenbahnen, Banken und belebte öffentliche Plätze. Beispiele für den Einsatz im nicht-öffentlichen Sektor finden sich bei Betriebsgeländen und privaten Gebäuden. Abbildung 2.1 stellt schematisch die Erstellungskette eines Überwachungsvideos dar. Die aufgenommenen Videodaten werden an Überwachungsmonitore gesendet und im Alarmfall zur Beweissicherung gespeichert.

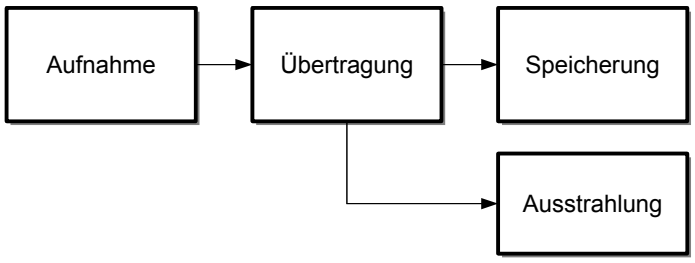


Abbildung 2.1: Erstellungskette eines Überwachungsvideos

Aktuelle Videoüberwachungssysteme speichern die Kamerabilder laut Gückel und Zumpe [Pro08] in den folgenden Formaten:

- Motion-JPEG für hoch-auflösende Bilder und schnelle Kameraschwenks (ca. 3 Bilder pro Sekunde)

- MPEG-4 für Netzwerke mit geringer Bandbreite

Darüber hinaus wird über die Einführung von H.264 als zukünftiges Videoformat diskutiert. Bevor dieses Format jedoch eingesetzt werden kann ist laut Gückel und Zumpe [Pro08] eine Standardisierung dieses Formats für Videoüberwachungssysteme erforderlich. Die Übertragung der Kamerabilder erfolgt in der Regel analog über Videokreuzschienen oder digital mittels Switches oder WLAN. Aus Datenschutzgründen gehen wir davon aus, dass Aufnahmen nur in Alarmsituationen für längere Zeit gespeichert werden und sonst bereits nach kurzer Zeit gelöscht werden.

In Videoüberwachungssystemen muss an verschiedenen Stellen ein Schutz gegen Angriffe eingebaut werden. Moderne Systeme können laut Orth [Pro08] eigenständig Manipulationen an der Kamera durch eine automatische Analyse der Kamerabilder erkennen. Selbstlernende Videodetektionssysteme teilen dabei die Gesamtszene in Vorder- und Hintergrundbilder auf. Aktuell eintreffende Bilder von der Kamera werden anschließend mit den gespeicherten Vorder- und Hintergrundmustern verglichen. Verändern sich die Inhalte der Kamerabilder stärker als zugelassen, so kann dies ein Hinweis auf eine Alarmsituation sein. Auslöser kann ein Defekt der Kamera als auch deren Verstellen oder Verdecken sein.

Um Man-in-the-middle-Angriffe bei der digitalen Übertragung abzuwehren werden laut Orth [Pro08] Verschlüsselungsmechanismen eingesetzt, z.B. HTTPS oder WPA-Verschlüsselung. Die Signale können dadurch auf dem Übertragungsweg nicht unbemerkt verändert werden.

Ein wichtiger Aspekt für die Gerichtsverwertbarkeit ist der Nachweis der Unversehrtheit der Videodaten nach ihrer Speicherung. Besteht die Möglichkeit, dass die Daten nach der Speicherung manipuliert wurden, ist das Beweismaterial nicht mehr glaubwürdig. Neben kryptografischen Technologien (z.B. Hashfunktionen, digitale Signatur [Sch96]) bieten auch digitale Videowasserzeichen eine Möglichkeit Manipulationen nachzuweisen.

Da nur in Alarmsituationen die Videodaten zur Beweissicherung langzeitgesichert werden ist der Anteil der gespeicherten Daten gering. Wir gehen daher nicht von einer zwischenzeitlichen Formatkonvertierung aus Speicherplatzgründen aus. Das eingesetzte Videowasserzeichen, das den Nachweis der Unversehrtheit erbringen soll, muss daher formatunabhängig und robust gegenüber verlustbehafteter Kompression sein, da es in verschiedenen Videoformaten eingesetzt werden kann.

Das Merkmal, das den Inhalt der Kamerabilder verifiziert, kann auf zwei verschiedene Arten arbeiten. In Abhängigkeit der aufgenommenen Bilder kann entweder der Schutz der gesamten Szene oder nur einzelner Objekte notwendig sein. Werden Demonstrationen oder belebte Plätze aufgenommen, so ist jedes Detail zur Beweissicherung wichtig. Handelt es sich im anderen Fall um statische Szenen (bspw. um geschlossene Räume) ist möglicherweise nur der Schutz der Vordergrundinformationen (bspw. Personen, bewegliche Objekte, eingeblendete Uhrzeit) notwendig. Neben den einzelnen Bildern muss deren zeitliche Abfolge geschützt werden. Werden Szenen

vertauscht, eingefügt oder entfernt so kann dies die Gesamtaussage der Aufzeichnung verändern. Damit ist sie als Beweismittel nicht mehr verwendbar. Darüber hinaus darf eine Formatkonvertierung und verlustbehaftete Komression bei gleichbleibender Qualität das Merkmal nicht beeinträchtigen.

Der Einsatz von Videowasserzeichen würde sich in diesem Szenario gut eignen, da die Informationen über den ursprünglichen Inhalt nicht separat gespeichert werden müssten um später einen Vergleich durchzuführen. Das Verfahren würde die Inhalts-Informationen direkt einbetten und würde auch nach verlustbehafteter Kompression und Formatumwandlung noch im Video enthalten sein, was bei einer angehängten Signatur nicht der Fall sein würde.

2.2 Nachrichtensendungen

Videos in Nachrichtensendungen sind ein wichtiges Mittel um Informationen über bestimmte Sachverhalte und Ereignisse zu vermitteln. Dazu gehören u.a. Berichte aus Kriegsgebieten und von Tatorten. Zu Nachrichtenvideos zählen wir zusätzlich Aufnahmen von politischen Reden. Abbildung 2.2 stellt schematisch die Erstellungskette eines Nachrichtenvideos dar.

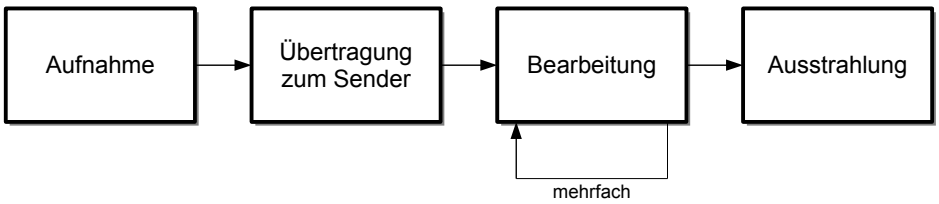


Abbildung 2.2: Erstellungskette eines Nachrichtenvideos

Nach der Aufnahme wird das Video an den Sender übertragen. Dies kann über Internet, Satellit oder direkt beim Sender geschehen. Bei der anschließenden Bearbeitung kommen verschiedene Bildverarbeitungsschritte zum Einsatz, z.B. die Veränderung des Kontrastes. Szenen können aufgrund der Sendedauer aus dem Video geschnitten werden. Um Details hervorzuheben, kann eine Ausschnittbildung vorgenommen werden. Darüber hinaus werden das Senderlogo und zusätzliche Informationen eingeblendet, z.B. der Aufnahmeort, der Videokontext oder der Name des Reporters. Das Nachrichtenvideo kann über verschiedene Wege ausgestrahlt werden, z.B. über Kabel, Satellit, DVB-T und Internet.

Durch die ausgestrahlten Nachrichtenvideos kann die Meinungsbildung der Zuschauer beeinflusst werden. Dies ist insbesondere dann von Bedeutung, wenn die Öffentlichkeit für eine bestimmte politische Entscheidung gewonnen werden soll (bspw. für einen militärischen Einsatz). In diesem Fall kann modifiziertes Videomaterial diese Meinungsbildung in eine bestimmte Richtung lenken. Das Heraustrennen von Szenen aus dem Gesamtvideo kann dessen Gesamtaussage verändern. Eine solche Manipulation erfolgte durch den US-TV-Sender Fox im Mai 2009 nach einer Anhörung

des ehemaligen US-Vizepräsidenten Al Gore zu seinen geschäftlichen Aktivitäten.[1] Gore bestätigte während der Anhörung zum Thema Emissionsrechtehandel, dass er Partner einer Firma sei, die durch Emissionsrechtehandel Geld verdient. Gore selbst erzielte durch seine Partnerschaft hohe Einnahmen, die er jedoch einer gemeinnützigen Organisation gespendet hatte. Die letzte Aussage über die Spenden fehlt in der durch Fox ausgestrahlten Version, so dass der Eindruck entstehen musste, dass Gore sich durch seine Umweltschutzaktivitäten persönlich bereichern würde.

Eine weitere Möglichkeit die öffentliche Meinung zu beeinflussen ist die Ausstrahlung politischer Reden und Interviews. Aussagen, die aus dem Zusammenhang herausgetrennt werden, können die Meinung über einen Politiker negativ beeinflussen und dessen Ansehen beschädigen und seine Glaubwürdigkeit beeinträchtigen.

Digitale Wasserzeichen eignen sich in diesem Anwendungsszenario um die Glaubwürdigkeit des ausgestrahlten Nachrichtenbeitrags zu überprüfen. Da der Großteil der Nachrichtensendungen mittlerweile im Internet verfügbar ist besteht die Möglichkeit ein Integritätswasserzeichen online verifizieren zu lassen. Interessierte Nutzer könnten dann anhand des eingebetteten Wasserzeichens überprüfen, an welchen Stellen das ursprüngliche Material bearbeitet wurde. Dazu ist es notwendig, dass das Wasserzeichen bereits bei der Aufnahme direkt in der Kamera eingebettet wird. Der Kameramann hat in diesem Fall keinen Einfluss auf den Einbettungsprozess. Ein erster Ansatz für eine solche „vertrauenswürdige Kamera“ wurde bereits 1993 in [Fri93] diskutiert. Dabei generiert die Kamera neben dem eigentlichen Bild intern dessen digitale Signatur. Der Schlüssel zur Erzeugung der Signatur wird in einem internen Mikroprozessor gespeichert. Neben der Beschreibung des Videoinhalts können Zusatzinformation wie die GPS-Koordinaten des Aufnahmeortes und der Aufnahmezeitpunkt als Wasserzeichennachricht eingebettet werden, um weitere Anhaltspunkte für die Glaubwürdigkeit des Nachrichtenbeitrages zu erhalten.

Das Wasserzeichen muss zuverlässig die Informationen über den Videoinhalt über die verschiedenen Wege und durch die verschiedenen Instanzen der Erstellungskette transportieren. Dazu gehört die Robustheit gegenüber verlustbehafteter Kompression und Formatkonvertierung bei der Übertragung zum ausstrahlenden Sender. Dabei kann die Stärke der verlustbehafteten Kompression in Abhängigkeit der verfügbaren Bandbreite zur Übermittlung stark variieren. Eine Übertragung via Satellit verfügt über die geringste Bandbreite während beim direkten Einspielen beim ausstrahlenden Sender von keinem Qualitätsverlust ausgegangen werden muss. Bei der Ausstrahlung des Nachrichtenbeitrags ist ebenfalls in Abhängigkeit des Kanals (Internet, DVB-T, Kabel) von einer verlustbehafeten Kompression und Formatkonvertierung auszugehen. Neben der Robustheit gegenüber den genannten Modifikationen muss das Wasserzeichen zusätzlich robust gegenüber Kontrastveränderung, Ausschnittbildung und dem Einbringen kleiner Logos und Untertitel sein, um den Videoinhalt auch noch nach der Bearbeitung verifizieren zu können.

Das Merkmal sollte auf zwei Wegen den visuellen Inhalt verifizieren können. Im ersten Teil sollte der Gesamtinhalt eines Videobildes geschützt werden. Dazu eig-

[1] http://mediamatters.org/research/200905010049 (Aufruf am 21.05.2011)

nen sich beispielsweise robuste Hashunktionen [OKH01]. Nach der Bearbeitung im ausstrahlenden Sender würde dieses Merkmal jedoch nicht mehr den Videoinhalt verifizieren. Daher ist der Einsatz eines zweiten Merkmals notwendig, das wichtige Bestandteile des Videos verifiziert, die auch nach Bearbeitung des Videos für die Gesamtaussage notwendig sind. Darüber hinaus muss das Merkmal die zeitliche Abfolge der Bilder verifizieren. Schnitte und Veränderungen in der Reihenfolge der Videosequenzen können die Gesamtaussage verändern und müssen somit erkannt werden können. Wie auch beim zugrunde liegenden Wasserzeichen muss das Merkmal robust gegenüber verlustbehafteter Kompression und Formatkonvertierung sein.

Die Anreicherung von Live-Aufnahmen mit Archiv-Material wurde in der jüngsten Vergangenheit Thema einer lebhaften Debatte über Manipulationen der Presse. Während der Fußball-Europameisterschaft 2012 in Polen und der Ukraine (Euro 2012) wurde während der Begegnung Niederlande gegen Deutschland eine Szene eingespielt, die den Bundestrainer Joachim Löw zeigt, wie er einem Balljungen aus Spaß den Ball entwendet. Da die Aufnahmen während des Spiels gezeigt wurden, wurde beim Zuschauer der Eindruck erzeugt, dass Löw so ruhig ist, dass er sich während des Spiels einen Spaß mit dem Balljungen erlaubt. Dass die Aufnahmen vor dem Spiel gemacht wurden, war nicht gekennzeichnet. Diese Manipulation an der Zeitachse und weitere Manipulationen während der Euro 2012 lösten eine Debatte aus, inwiefern die Presse durch gezielte Manipulationen bereits heute manipuliert wird.

2.3 Historische Aufnahmen

Historische Dokumente werden allgemein als das kulturelle Gut einer Nation angesehen. Gehen sie verloren (bspw. durch einen Brand) so geht damit ein Stück Geschichte verloren. Daher ist es wichtig sie mit modener Technik zu digitalisieren um deren Inhalt vor dem Verfall oder anderer Zerstörung zu schützen. Durch die Digitalisierung ist eine einfache Vervielfältigung möglich. Dies hat zum Effekt, dass wertvolle historische Dokumente einer breiten Öffentlichkeit zugänglich gemacht werden können. Abbildung 2.3 zeigt die Erstellungskette für historische Aufnahmen.

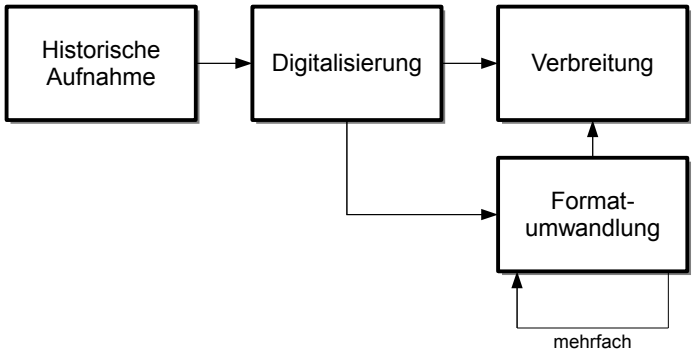


Abbildung 2.3: Erstellungskette für historische Aufnahmen

Wir gehen davon aus, dass die digitalen Kopien historischer Dokumente für lange Zeit gespeichert werden. Über diesen Zeitraum hinweg kann mehrfach eine Formatumwandlung stattfinden, um Speicherplatz zu sparen. Dabei findet möglicherweise eine verlustbehaftete Kompression statt. Um Historikern auch mit dem digitalisierten Material Analysen weiterhin zu ermöglichen gehen wir davon aus, dass die Kompression mit hoher Qualität durchgeführt wird.

Historische Aufnahmen waren schon in der Vergangenheit Ziel von Manipulationen. So versuchte bereits Josef Stalin seinen politischen Gegner Leo Trotzki aus offiziellen Dokumenten mittels Bildmanipulation löschen zu lassen, um ihn aus dem öffentlichen Gedächtnis zu entfernen [Sti04]. Dadurch sollte die Erinnerung an historische Ereignisse, wie die Rede Lenins in Moskau, manipuliert werden. Sollen historische Dokumente einer breiten Öffentlichkeit zugänglich gemacht werden, so muss die Möglichkeit bestehen deren Unversehrtheit nachzuweisen. Andernfalls können sie dazu verwendet werden, um geschichtliche Sachverhalte in einem neuen Kontext darzustellen. Digitale Wasserzeichen eignen sich hierfür, da keine zusätzlichen Informationen gespeichert werden müssen. Das Dokument kann beispielsweise durch das ausgebende Archiv verifiziert werden.

Das verifizierende Wasserzeichenverfahren darf die visuelle Qualität der Aufnahme nicht beeinträchtigen, um Analysen nicht zu verfälschen. Hierin sehen wir eine große Herausforderung. Neben den Merkmalsinformationen kann das Wasserzeichen eine ID enthalten, die zu zusätzlichen Informationen führt. Diese ID kann beispielsweise zu einem Datenbankeintrag führen, der einen Zeitstempel, den Aufnahmeort und Informationen über den dargestellten Inhalt (Annotationen) enthält.

Das inhaltsbeschreibende Merkmal muss zwei Funktionen erfüllen. Wir gehen davon aus, dass für historische Analysen potentiell der gesamte Inhalt geschützt werden muss. Das bedeutet also, dass das Merkmal den gesamten Inhalt beschreiben muss. Darüber hinaus ist es wichtig, dass das Merkmal die zeitliche Abfolge der Bilder widerspiegelt. Werden Szenen aus der Aufnahme entfernt, so muss das Entfernen bei der Verifikation des Videos erkannt werden.

2.4 Weitere mögliche Anwendungsszenarien

Die oben beschriebenen Anwendungsszenarien stellen nur einen kleinen Teil von möglichen Anwendungsgebieten dar. In dieser Arbeit beschränken wir uns auf diese drei Szenarien, da sie nach unserer Auffassung vollständig das Spektrum der Anforderungen an die inhaltsbeschreibenden Merkmale und deren zugrunde liegenden robusten Videowasserzeichen abdecken. Weitere mögliche Anwendungsszenarien für Integritätswasserzeichen können u.a. sein:

- Videos, aufgenommen mit mobilen Endgeräten: Dieses Szenario ist vergleichbar mit dem Videoüberwachungsszenario und Nachrichtenszenario. Aufnah-

men mit mobilen Endgeräten können einerseits die öffentliche Meinung beeinflussen und andererseits als Beweismittel verwendet werden.

- Aufnahmen von Materialtests: Werden Aufnahmen von misslungenen Crash-Tests und Materialtests öffentlich, so können sie den Ruf einer Firma negativ beeinflussen. Um dies zu verhindern könnten solche Aufnahmen manipuliert werden, um die Firma in der öffentlichen Meinung in einem besseren Licht dastehen zu lassen.

- Medizinische Aufnahmen: Gerade in diesem Bereich sollte sichergestellt sein, dass die Aufnahmen nicht versehentlich verfälscht wurden. Hierbei können auch invertierbare Wasserzeichenverfahren zum Einsatz kommen [Dit00].

- Aufnahmen von Polizeiverhören: Auch hier sollte der Inhalt authentisch sein, da solche Aufnahmen als gerichtsverwertbare Beweise verwendet werden könnten.

2.5 Zusammenfassung

In der folgenden Tabelle fassen wir noch einmal die Anforderungen an das inhaltsbeschreibende Merkmal in Abhängigkeit der Anwendungsszenarios zusammen (siehe Tabelle 2.1). Während wir für historische Aufnahmen eine vollständige Verifikation des Inhalts und der Zeitachse empfehlen, kommt bei der Videoüberwachung und bei den Nachrichtensendungen auch eine Verifikation von einzelnen Objekten zusätzlich zur Zeitachse in Frage.

Szenario	Vollständig	Objekt	Zeitlich
Videoüberwachung	X	X	X
Nachrichtensendungen	X	X	X
Historische Aufnahmen	X	-	X

Tabelle 2.1: Verifikationsanforderungen an das inhaltsbeschreibende Merkmal

Kapitel 3

Stand der Technik

Dieses Kapitel stellt den Stand der Technik zu Beginn unserer Forschung dar. Es untergliedert sich in zwei Abschnitte: Robuste Videowasserzeichen und Integritätswasserzeichen. Letzteres behandelt zudem separat fragile und semifragile Bild- und Videowasserzeichen.

3.1 Robuste Videowasserzeichen

In diesem Abschnitt stellen wir zwei robuste Videowasserzeichenverfahren vor, die bei der Entwicklung für die Basis unserer Integritätswasserzeichen maßgeblich von Bedeutung waren. Ausschlaggebend für die Wahl dieser Verfahren waren deren hohe Robustheit, die gute Transparenz und eine ausreichend hohe Kapazität. Beide Verfahren arbeiten auf Basis der Diskreten Kosinus Transformation (DCT).

3.1.1 Additives Verfahren nach Fridrich

In [Fri97] stellt Fridrich ein Wasserzeichenverfahren vor, welches ein nicht wahrnehmbares Rauschmuster auf ein Bild aufträgt. Das Muster, welches seine höchste Energie in den niedrigen Frequenzen enthält, wird mittels eines zellulären Automaten erzeugt. Wir beschreiben im Folgenden die Mustergenerierung, den Einbettungs- und Ausleseprozess des Verfahrens. Für das ursprüngliche Verfahren wird im Ausleseprozess das Original benötigt. Die Weiterentwicklung zu einem blinden Verfahren wird in den folgenden Unterabschnitten vorgestellt.

Die Mustergenerierung beginnt mit der Erzeugung eines pseudo-zufälligen binären Rauschmusters, das zu gleichen Teilen Nullen und Einsen enthält. Die Größe $m \times n$ des Musters entspricht der Auflösung des zu markierenden Bildes. Es wird also ein Muster pro Bild aufgetragen, das einem Präsenz-Wasserzeichen entspricht. Die Erzeugung wird mittels eines geheimen Schlüssels gesteuert. Fridrich schlägt vor hierfür

den Hashwert des Originalbildes in Kombination mit einer Autoren-ID zu verwenden. Der Autor kann sich dadurch als Urheber des Bildes identifizieren, da er sowohl die richtige Autoren-ID weiß als auch im Besitz des Originalbildes ist und damit das Wasserzeichen wieder auslesen kann. Da das entstandene Rauschmuster viele hohe Frequenzen enthält und damit gegen verlustbehaftete Kompression eine geringere Robustheit aufweist, wird im nächsten Schritt ein zellulärer Automat mit einer einfachen Entscheidungsregel verwendet. Für jedes Pixel p_i des Rauschmusters P wird die Summe seiner Nachbarn innerhalb einer 3×3 Nachbarschaft N berechnet. Ist die Bedingung $N < 5$ erfüllt, so wird p_i im Ergebnismuster auf 0 gesetzt, andernfalls auf 1. Der zelluläre Automat wird nach einer vorher definierten Anzahl von Wiederholungen gestoppt. Fridrich weist in [Fri97] darauf hin, dass bei einem Muster der Größe 128×128 Pixel weniger als 40 Wiederholungen benötigt wurden. Im letzten Schritt wird das Muster weichgezeichnet und seine Farbtiefe auf 16 reduziert. Anschließend werden die Pixel in den Intervall $[-8, 7]$ verschoben. Dadurch wird gewährleistet, dass die Pixel des zu markierenden Bildes um maximal 8 Einheiten verändert werden. Um das Bild X zu markieren wird das Muster P auf die Pixel des Bildes addiert:

$$X' = X + P \tag{3.1}$$

Der Ausleseprozess findet nach [Fri97] im Frequenzraum statt. Dazu werden das Originalbild X, das markierte Bild X' und das zu untersuchende Bild X^* mittels DCT in den Frequenzraum transformiert. Mit der Funktion $sim(X^*, X')$ wird über das Vorhandensein des Wasserzeichens entschieden:

$$sim(X^*, X') = \frac{D^* \cdot D}{\sqrt{D^* \cdot D^*}\sqrt{D \cdot D}} \tag{3.2}$$

In Gleichung (3.2) gilt $D = Y' - Y$ und $D^* = Y^* - Y$, wobei Y, Y' und Y^* die 1024 niedrigsten DCT-Koeffizienten der transformierten Bilder X, X' und X^* repräsentieren.

Die Experimente in [Fri97] zeigen, dass das Verfahren robust gegenüber verschiedenen Bildverarbeitungsoperationen, wie verlustbehaftete Kompression, Skalierung, Ausschnittbildung und Weichzeichnen ist. Dabei schwankt die Korrelation zwischen $0,38$ und $0,97$. Wird versucht das Wasserzeichen aus einem Bild auszulesen, das mit einem anderen Schlüssel generiert wurde, so beträgt die Korrelation maximal $0,175$. Durch diesen Schwellwert kann festgestellt werden, ob ein entsprechendes Wasserzeichen vorhanden ist oder nicht.

Erweiterung nach Dittmann

In [Dit00] stellt Dittmann eine Erweiterung des Verfahrens von Fridrich vor. Dabei wird es zu einem blinden Verfahren erweitert und die Möglichkeit zum Einbetten mehrerer Wasserzeichenmuster pro Bild eingeführt. Dadurch ist es möglich mehrere Bits einer Wasserzeichennachricht einzubetten, die beispielsweise Urheberinformationen oder Informationen über den Bild- oder Videoinhalt repräsentieren.

Zum Einbetten einer Wasserzeichennachricht wird das Bild in Blöcke B_j unterteilt. Der Zufallszahlengenerator wird mit einem geheimen Schlüssel initialisiert. Da das Verfahren im Ausleseprozess blind sein soll kann der Hashwert des Originalbildes nicht für die Schlüsselerstellung verwendet werden. Ein binäres Muster P_j, welches ein Bit m_j der Wasserzeichennachricht M repräsentiert, wird pseudo-zufällig erzeugt. Anschließend wird der in [Fri97] erwähnte zelluläre Automat dreimal auf das Muster P_j angewandt. Im nächsten Schritt wird das Muster verstärkt. Dazu werden alle Musterpositionen $p_{ji} \in P_j$, die eine 1 enthalten mit einer 3 belegt. Alle p_{ji}, die eine 0 enthalten, werden mit einer -3 belegt. Zum Schluß wird ein Mittelwertfilter der Größe 3×3 angewandt, um die Transparenz des Musters zu erhöhen. In Abhängigkeit des einzubettenden Bits m_j wird ein Block B_j mit dem Muster P_j und einem Verstärkungsfaktor α folgendermaßen markiert:

$$B'_j = \begin{cases} B_j + \alpha \cdot P_j & wenn \quad m_j = 1 \\ B_j - \alpha \cdot P_j & wenn \quad m_j = 0 \end{cases} \tag{3.3}$$

Im Ausleseprozess wird das eingebettete Muster P_j mittels Korrelation aus einem markierten und eventuell modifizierten Block B^*_j ausgelesen. Dazu werden aus den Pixeln von B^*_j zwei Mittelwerte gebildet:

- M_+ ist der Mittelwert aller Positionen b^*_{ji} in B^*_j, deren entsprechende Musterpositionen p_{ji} positiv sind.
- M_- ist der Mittelwert aller Positionen b^*_{ji} in B^*_j, deren entsprechende Musterpositionen p_{ji} negativ sind.

Ist die Bedingung $M_+ \geq M_-$ erfüllt, so wird aus B^*_j eine 1 ausgelesen, andernfalls eine 0.

Zur Verbesserung der Transparenz verwendet Dittmann ein visuelles Modell, das die Struktur der zu markierenden Blöcke analysiert. Enthält ein Block viele Kanten, so kann er gut markiert werden. Enthält ein Block viele Flächen, so ist er zur Markierung weniger geeignet, da das Muster leichter sichtbar werden würde. Dittmann weist darauf hin, dass durch die Verwendung des visuellen Modells die Robustheit des Verfahrens reduziert werden könnte, da auch nicht-markierte Blöcke analysiert werden würden. Um dies zu verhindern ist die Anwendung des visuellen Modells auch im Ausleseprozess notwendig.

Dieses Verfahren wird in Kapitel 5 modifiziert und bildet anschließend die Grundlage für unser Konzept, da es sich durch die Anwendung in der Praxis mit seiner guten Robustheit und hohen Kapazität bei guter Transparenz bewährt hat.

3.1.2 Energiedifferenz-Verfahren nach Langelaar et al.

Langelaar et al. stellen in [LLB99] ein Wasserzeichenverfahren für Bild- und Videodaten vor, das durch Erzwingen von Energiedifferenzen in Blockgruppen Wasserzeichennachrichten einbringt. Es ist robust gegenüber verlustbehafteter Kompression und weist durch die gezielte Manipulation der hohen Frequenzen eine gute Transparenz auf. Wir stellen dieses Verfahren hier vor, da es die Grundlage für das in Abschnitt 6.1 vorgestellte inhaltsfragile Wasserzeichenverfahren bildet.

Um eine Wasserzeichennachricht in ein Bild oder Videoframe einzubetten muss es zunächst in Blöcke der Größe 8×8 Pixel unterteilt werden. Anschließend werden die Blöcke in den Frequenzraum mittels DCT transformiert und auf Basis eines geheimen Schlüssels gemischt. Die DCT-Blöcke werden dann in Gruppen der Größe n unterteilt. Jede Gruppe wird nochmals in eine Untergruppe A und Untergruppe B der Größe $n/2$ unterteilt. Um ein Bit m_j der Wasserzeichennachricht M einzubetten wird zwischen der Untergruppe A_j und B_j der Gruppe G_j eine Energiedifferenz erzwungen. Die Methode zum Erzwingen der Energiedifferenz beschreiben wir im Folgenden.

Die Energie einer Untergruppe wird über alle DCT-Koeffizienten der Untermenge $S(c)$ berechnet:

$$S(c) = \{i \in \{0, 63\} \,|(i > c)\}\,, \tag{3.4}$$

wobei c einen so genannten Cut-Off-Point bezeichnet, dessen Rolle später erklärt wird. Die Energie wird folgendermaßen berechnet:

$$E_{A,B}(c, n) = \sum_{b=0}^{n/2-1} \sum_{i \in S(c)} \theta_{b,i}^2\,, \tag{3.5}$$

wobei $\theta_{b,i}$ hier den DCT-Koeffizienten i (in Zick-Zack-Anordnung) des Blockes b der Untergruppe A oder B bezeichnet. Die Energiedifferenz $D(c, n)$ zwischen den Untergruppen ist folgendermaßen definiert:

$$D(c, n) = E_A(c, n) - E_B(c, n)\,. \tag{3.6}$$

Um ein Bit m_j der Nachricht M einzubetten muss folgende Bedingung erfüllt werden:

$$D(c,n) = \begin{cases} > 0 & falls \quad m_j = 0 \\ < 0 & falls \quad m_j = 1 \end{cases} \quad . \tag{3.7}$$

Gilt $m_j = 0$ so müssen in Untergruppe B alle DCT-Koeffizienten nach einem berechneten Cut-Off-Point C gelöscht, d.h. auf 0 gesetzt werden. Gilt $m_j = 1$ so muss Untergruppe A in gleicher Weise manipuliert werden, um die Bedingung aus Gleichung (3.7) zu erfüllen. Die Berechnungsvorschrift von C ist durch die folgende Formel definiert:

$$C(n, D, c_{min}) = max \left\{ \begin{array}{l} c_{min} \\ max \left\{ i \in \{0, 63\} \,|(E_A(i,n) > D) \wedge (E_B(i,n) > D) \right\} \end{array} \right\} \tag{3.8}$$

c_{min} stellt in Gleichung (3.8) einen Schwellwert dar. Er hat maßgeblich Einfluss auf die Verfahrensparameter Transparenz und Robustheit. Ist c_{min} hoch, so ist das Wasserzeichen sehr transparent, kann jedoch eine geringere Robustheit zur Folge haben. Ist dagegen c_{min} niedrig, so ist das Wasserzeichen robust, kann jedoch sichtbar werden. Methoden zur Bestimmung des optimalen Schwellwerts c_{min} werden in [LLB99] diskutiert. Ein weiterer Parameter, der die Verfahrensparameter des Wasserzeichens beeinflusst ist die Differenz D. Je höher D ist desto besser kann das Wasserzeichen später auch nach Manipulationen wieder ausgelesen werden. Eine hohe erzwungene Energiedifferenz zieht jedoch eine höhere Anzahl zu löschender DCT-Koeffizienten nach sich, die sich negativ auf die Transparenz auswirken kann. Abbildung 3.1 demonstriert das Einbringen einer 0 in eine Gruppe G_j mit $n = 16$ Blöcken. In Untergruppe A werden die hohen Frequenzen bis zum Cut-Off-Point c eliminiert um die Differenz D zu erzwingen.

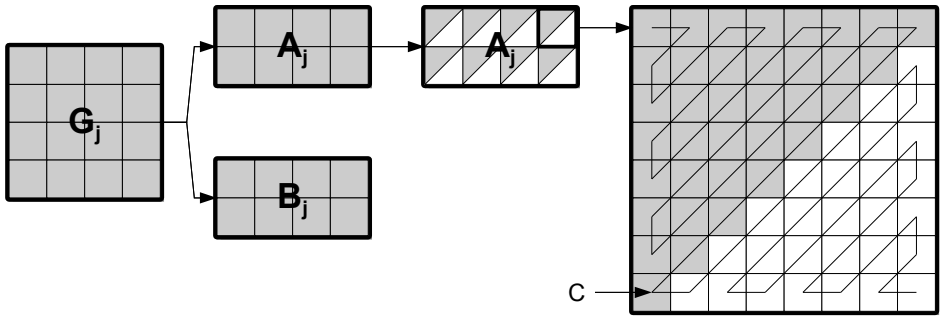


Abbildung 3.1: Markierung einer Gruppe nach Langelaar et al.

Im Ausleseprozess muss der Cut-Off-Point C aus den Untergruppen berechnet werden. Dazu wird folgende Formel angewendet:

$$C(n, D') = min \left\{ C_A(n, D'), C_B(n, D') \right\} \tag{3.9}$$

mit $C_{A,B}(n, D') = min\{i \in \{0, 63\} \,|E_{A,B}(i, n) < D'\}$. Anschließend kann C in (3.6) eingesetzt werden. Ist $D < 0$ so gilt $m_j = 1$. Analog dazu gilt $m_j = 0$, falls $D > 0$.

3.2 Integritätswasserzeichen

In diesem Abschnitt stellen wir verschiedene Verfahren zum Schutz der Integrität von Bildern und Videos vor, die in der Literatur veröffentlicht wurden. Dabei gehen wir zunächst auf Eigenschaften und mögliche Angriffe auf Integritätswasserzeichen ein und beschäftigen uns dann mit fragilen und semifragilen Bild- und Videowasserzeichen.

3.2.1 Eigenschaften

Nach Lin et al. [LC00] kann ein Wasserzeichen zum Erkennen von Manipulationen anhand folgender Eigenschaften bewertet werden:

- Sensitivität: Das Verfahren ist sensitiv gegenüber Manipulationen, die den Inhalt des Bildes/Videoframes verändern. Beispiele hierfür sind das Entfernen und das Einfügen von Objekten oder Videoframes.
- Robustheit: Das Verfahren ist robust gegenüber Manipulationen, die den Inhalt des Bildes/Videoframes nicht verändern. Beispiele hierfür sind verlustbehaftete Kompression und Formatumwandlung.
- Sicherheit: Das Wasserzeichen kann nicht gefälscht werden ohne dass diese Veränderungen bemerkt werden.
- Lokalisierbarkeit: Das Verfahren kann die Stellen aufzeigen, an denen der Inhalt verändert wurde.
- Wiederherstellbarkeit: Das Verfahren bietet die Möglichkeit veränderten Inhalt wieder in seinen Ursprungszustand zu versetzen oder den Original-Inhalt in reduzierter Qualität anzuzeigen.

Fragile Wasserzeichenverfahren zeigen bereits minimale Veränderungen am Material an. Dies kann bedeuten, dass bereits ein veränderter Pixelwert erkannt wird. Sie verfügen über eine sehr hohe Sensitivität bei nicht vorhandener Robustheit. Dem gegenüber zeigen semifragile Wasserzeichenverfahren Veränderungen nur an, wenn sie den Inhalt verändern bzw. einen vorher definierten Grad an Veränderungen überschreiten. Welche Veränderungen in welcher Intensität erlaubt sind hängt dabei vom jeweiligen Anwendungsszenario ab.

3.2.2 Angriffe

Bei der Entwicklung eines Wasserzeichenverfahrens spielt dessen Sicherheit eine wichtige Rolle. Ist es einem Angreifer möglich ein gefälschtes Bild als authentisch verifizieren zu lassen, so ist das Verfahren unbrauchbar. Fridrich stellt in [Fri02] verschiedene Methoden vor, die die Sicherheit von Bildwasserzeichenverfahren angreifen:

1. Stego-Image Attack: Der Angreifer verfügt über ein einzelnes markiertes Bild. Ziel ist hier unerkannte Veränderungen einzubringen bzw. geheime Informationen über das Verfahren zu erhalten. Geheime Informationen können der Schlüssel zur Einbettung des Wasserzeichens oder Informationen über Parametereinstellungen des Verfahrens sein.

2. Multiple Stego-Image Attack: Bei diesem Angriff verfügt der Angreifer über mehrere markierte Bilder. Wiederum ist das Ziel unerkannte Veränderungen einzubringen bzw. geheime Informationen zu erhalten.

3. Verification Device Attack: Der Angreifer verfügt über die Möglichkeit ein markiertes Bild zu verifizieren. Dabei kann die Rückmeldung entweder binärer Art (Bild ist authentisch / nicht authentisch) oder ein Bitmap sein, das die Positionen in dem Bild markiert, die nicht authentisch sind. Auch hier wird das Ziel verfolgt, unerkannte Veränderungen einzubringen bzw. geheime Informationen zu erhalten.

4. Cover-Image Attack: Der Angreifer verfügt über mehrere Paare von markierten und zugehörigen unmarkierten Bildern. Das Ziel ist wiederum unerkannte Veränderungen einzubringen bzw. geheime Informationen zu erhalten.

5. Chosen Cover-Image Attack: Der Angreifer kann bei diesem Angriff seine eigenen Bilder markieren lassen. Ziel ist es hier Informationen über den geheimen Schlüssel zu erhalten.

Ein sehr populärer Angriff, der zu der Klasse der Multiple Stego-Image Attack gezählt werden kann wird von Holliman und Memon in [HM00] vorgestellt und ist in der Literatur als Holliman-Memon-Attacke bekannt geworden. Es wird dabei davon ausgegangen, dass der Angreifer über eine Datenbank von Bildern verfügt, die mit dem selben Schlüssel und dem selben Wasserzeichenmuster markiert wurden. Ein Angreifer kann dann ein beliebiges Bild markieren, indem er für jeden Block des unmarkierten Bildes die Datenbank nach Blöcken an der selben Position durchsucht und durch den ähnlichsten markierten Block ersetzt. Das so entstandene Bild, das dem unmarkierten Bild sehr ähnlich sieht, wird vom Verfahren als authentisch verifiziert, wenn die Blöcke nicht untereinander in Beziehung gebracht werden.

Es gilt zu beachten, dass die genannten Angriffe auch auf Videowasserzeichen anwendbar sind.

3.2.3 Fragile Wasserzeichen für Bilder

Verfahren von Yeung und Mintzer

Eines der ersten fragilen Wasserzeichenverfahren für Bilddaten wurde von Yeung und Mintzer in [YM97] vorgestellt. Dazu wird das Wasserzeichenmuster $W(i,j)$ in ein binäres Muster $b(i,j)$ überführt und in das zu markierende Bild $I(i,j)$ eingebracht um das markierte Bild $I'(i,j)$ zu erhalten. In Abhängigkeit eines geheimen Schlüssels wird die Funktion $WX(*)$ aufgerufen, die das binäre Wasserzeichenmuster aus $I(i,j)$ extrahiert. Sie ist für RGB-Werte nach Gleichung (3.10) und für Helligkeitswerte nach Gleichung (3.11) definiert.

$$WX(*) : b(i,j) = LUT_R(I_R(i,j)) \oplus LUT_G(I_G(i,j)) \oplus LUT_B(I_B(i,j)) \quad (3.10)$$

$$WX(*) : b(i,j) = LUT(I(i,j)) \quad (3.11)$$

$LUT()$ beschreibt dabei eine mit einem geheimen Schlüssel initialisierte Tabelle, die für jeden Farb- bzw. Helligkeitswert eines Bildes einen binären Wert enthält und $\oplus$ den binären Operator „Exklusives Oder“ (XOR). Das Verfahren von Yeung und Mintzer modifiziert die Werte von $I(i,j)$ so lange, bis die Gleichungen (3.10) bzw. (3.11) für alle $I'(i,j)$ erfüllt sind. Um die durchschnittlichen Grau- bzw. Farbwerte zu erhalten verwenden die Autoren einen speziellen Prozess, der den eingebrachten Fehler (d.h. die Veränderung der Pixelwerte) über benachbarte Pixel streut. Zum Einbringen des Wasserzeichenmusters werden die 5 niedrigsten Bitebenen modifiziert.

Im Verifikationsprozess wird die Funktion $WX(*)$ auf das markierte und möglicherweise modifizierte Bild $I'^*(i,j)$ angewandt um das binäre Wasserzeichenmuster $b'^*(i,j)$ zu erhalten. Durch einen Vergleich zum bekannten Wasserzeichenmuster $b(i,j)$ können modifizierte Stellen lokalisiert und markiert werden.

Das Verfahren von Yeung und Mintzer verursacht durch die geringe Modifikation der Pixelwerte keine sichtbaren Artefakte. Es verfügt über eine sehr hohe Lokalisierungsfähigkeit bei nicht vorhandener Robustheit. Die Sicherheit des Verfahrens basiert auf dem geheimen Schlüssel, der die Extraktionsfunktion $WX(*)$ steuert. Ohne diesen Schlüssel besteht nicht die Möglichkeit das Wasserzeichen zu detektieren. Fridrich et al. weisen jedoch in [FGM00] und [FGD02] nach, dass bei der Verwendung des selben Schlüssels und des selben Wasserzeichenmusters bereits mit zwei markierten Bildern 90% der Extraktionsfunktion $WX(*)$ ermittelt werden können. Das Verfahren von Yeung und Mintzer ist somit in seiner ursprünglichen Form anfällig gegen die Multiple Stego-Image Attack.

Verfahren von Wong

Ein weiteres Verfahren zum Einbringen fragiler Wasserzeichen in Bildern wurde von Wong in [Won98] vorgestellt. Dabei wird das zu markierende Bild in nichtüberlappende Blöcke der Größe $W \times H$ unterteilt. Der Markierungsprozess behandelt jeden Block separat. Das Verfahren sieht zwei Modi vor.

Im Private-Key-Modus wird aus den sieben höchsten Bitebenen (Most Significant Bit, MSB) eines 8-Bit Farbwertes ein Hashwert mit Hilfe einer schlüsselabhängigen Hashfunktion gebildet. Dieser Hashwert wird mit einem vorher definierten binären Wasserzeichenmuster mittels XOR-Operation verknüpft. Das Ergebnis der XOR-Operation ersetzt die niedrigste Bitebene (Least Significant Bit, LSB) des selben Blockes. Im Verifikationsprozess wird der Hashwert aus den sieben MSBs mit der selben Hashfunktion gebildet und mittels XOR-Operation mit den LSBs verknüpft. Das Ergebnis wird anschließend mit dem binären Wasserzeichenmuster verglichen, wobei Unterschiede auf manipulierte Blöcke hinweisen.

Im Public-Key-Modus wird aus den sieben MSBs mit einer schlüsselunabhängigen Hashfunktion ein Hashwert gebildet und dieser anschließend mittels XOR-Operation mit dem binären Wasserzeichenmuster verknüpft. Das Ergebnis wird mit einem Public-Key-Verschlüsselungsverfahren mit dem privaten Schlüssel verschlüsselt. Die verschlüsselte Information ersetzt wiederum die LSBs des selben Blockes. Im Verifikationsprozess wird der Hashwert der MSBs eines Blockes berechnet und mittels XOR-Operation mit den entschlüsselten LSBs verknüpft. Die Entschlüsselung erfolgt mit dem öffentlichen Schlüssel. Das Ergebnis wird mit dem binären Wasserzeichenmuster verglichen. Wiederum zeigen Unterschiede manipulierte Blöcke auf.

Das vorgestellte Verfahren weist eine sehr gute Transparenz aufgrund der Modifikation der LSBs auf. Gleichzeitig verfügt es über eine sehr gute Lokalisierbarkeit bei nicht vorhandener Robustheit. Allerdings ist das Verfahren ebenso wie das Verfahren von Yeung und Mintzer in seiner ursprünglichen Version anfällig gegen die Holliman-Memon-Attacke.

Verfahren von Fridrich

In [Fri02] stellt Fridrich ein Verfahren vor, das auf dem Verfahren von Wong aufbaut und sicher gegen alle in Abschnitt 3.2.2 vorgestellten Angriffe ist. Der Aufbau des Wasserzeichenmusters ist jetzt abhängig von den Positionsinformationen des zu schützenden Blockes. Der Aufbau eines Blockes der Größe 8×16 Pixel (128 Pixel) ist in Abbildung 3.2 dargestellt. Die ersten 52 Bits enthalten Informationen über die Blockposition und den Bild-Index. Diese 52 Bits werden wiederholt um später die Authentizität des Blockes unabhängig von seiner Position verifizieren zu können. Die letzten 24 Bits sind für weitere Informationen, wie die Kamera-ID und die ursprüngliche Bildauflösung reserviert. Der Einbettungsprozess und Verifikationsprozess erfolgt analog zu [Won98]. Ebenso verfügt das Verfahren über einen Private-Key-Modus und einen Public-Key-Modus. Durch das Verknüpfen des Bild-

Index mit den Positionsinformationen ist es einem Angreifer nun nicht mehr möglich ein unmarkiertes Bild aus verschiedenen markierten Bildern zu erstellen, da jedes Bild über ein individuelles Wasserzeichenmuster verfügt.

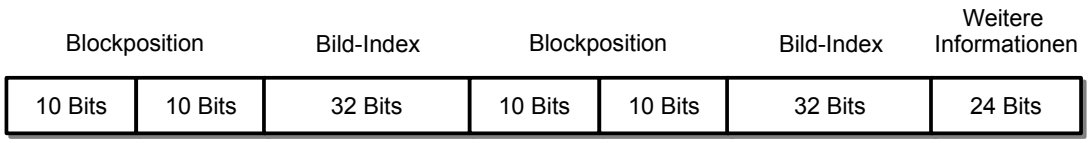


Abbildung 3.2: Aufbau des Wasserzeichenmusters nach Fridrich [Fri02]

3.2.4 Semifragile Wasserzeichen für Bilder

Verfahren von Lin und Chang

In [LC00] stellen Lin und Chang ein semifragiles Wasserzeichenverfahren für JPEG-komprimierte Bilder vor. Das Verfahren basiert auf einem Signaturverfahren, das die Autoren in [LC98] vorstellen. Die Signatur beruht auf der Beobachtung, dass das Größenverhältnis zweier DCT-Koeffizienten F_p und F_q, die sich an der selben Position zweier beliebiger 8×8 DCT-Blöcke p und q befinden, nach verlustbehafteter JPEG-Kompression gleich bleibt. Diese Beobachtung gilt auch für MPEG-komprimierte Videos. Das Verfahren bettet sowohl Authentifizierungsbits zur Manipulationserkennung als auch Bits zur Wiederherstellung des ursprünglichen Inhalts in reduzierter Qualität ein.

Zur Generierung der Authentifizierungsbits werden β_a Positionen in einem 8×8 DCT-Block ausgewählt. Die Positionen sind in der für JPEG und MPEG üblichen Zick-Zack-Reihenfolge angeordnet. Die Menge der ausgewählten Positionen B_p wird als Signatur-Generierungs-Bereich des Blockpaares (p, q) bezeichnet. Die Blockpaare werden durch eine geheime Blockzuordnungsfunktion $T_b : q = T_b(p)$ gebildet. Aus B_p werden die Authentifizierungsbits $\Phi_p(v), v \in B_p$ wie folgt berechnet:

$$\Phi_p(v) = \begin{cases} 1, & \Delta F_{p,q}(v) \geq 0 \\ 0, & \Delta F_{p,q}(v) < 0 \end{cases} \tag{3.12}$$

Für den Einbettungsprozess werden $\frac{1}{2}\beta_a$ weitere Koeffizienten jedes Blockes verwendet, die zur Menge E_p zusammengefasst werden. Der andere Teil der Authentifizierungsbits wird in den Block o eingebettet, der durch eine geheime Blockzuordnungsfunktion $T_a : o = T_a(p)$ ermittelt wird. Jeder zu markierende Koeffizient $F_{o,p}(v), v \in E_p$ wird dahingehend modifziert, dass das LSB des mit einem Qualitätsfaktor $Q'_m(v) = Q_m(v) + 1$ quantisierten Koeffzienten dem einzubettenden Authentifizierungsbit entspricht. Durch dieses Verfahren erlangt das Wasserzeichen Robustheit gegenüber verlustbehafteter JPEG-Kompression bis zu einem geheimen minimalen Qualitätsfaktor $Q_m(v)$.

Die Bitmenge Ψ wird zur Wiederherstellung des ursprünglichen Bildinhalts in reduzierter Qualität verwendet. Zur Bildung von Ψ wird die Auflösung des Bildes um den Faktor 2 reduziert. Anschließend wird auf dem skalierten Bild eine starke JPEG-Kompression mit anschließender Huffman-Kodierung durchgeführt. Die Bits eines 8×8 Blockes des reduzierten Bildes werden auf 4 Blöcke des zu markierenden Bildes verteilt und dort als Wasserzeichen eingebettet. Der Einbettungsprozess geschieht analog zum Einbettungsprozess der Authentifizierungsbits. Er wird durch den geheimen Qualitätsfaktor $Q_{mr}(v)$ und die geheime Blockzuordnungsfunktion T_r kontrolliert.

Im Verifikationsprozess werden zunächst die eingebetteten Authentifizierungsbits ausgelesen und mit den aktuellen Authentifizierungsbits verglichen. Werden Unterschiede festgestellt, so werden die Wiederherstellungsbits ausgelesen um den Ursprungsinhalt mit reduzierter Qualität wiederherzustellen.

Das Verfahren verfügt über eine gute Robustheit gegenüber JPEG-Kompression. Sie ermöglicht die Lokalisierung von Manipulationen und die Wiederherstellung des ursprünglichen Inhalts bei verminderter Qualität. Die Sicherheit wird durch die Geheimhaltung der Blockzuordnungsfunktionen, der Qualitätsfaktoren und der Koeffizientenmengen gewährleistet. Da die Nachricht abhängig vom Inhalt der Blockpaare ist, ist das Verfahren sicher gegenüber den in Abschnitt 3.2.2 vorgestellten Angriffen. Allerdings arbeitet das Verfahren nur auf JPEG-Bildern. Wie sich das Verfahren nach Formatkonvertierungen verhält wird aus [LC00] nicht ersichtlich.

Verfahren von Maeno et al.

In [MSCS06] verändern Maeno et al. das Verfahren von Lin und Chang dahingehend, dass bisher nicht erkannte Veränderungen besser detektiert werden können. Dazu wird der Signatur-Generierungsprozess um einen pseudo-zufälligen Wert erweitert und mehrere Teilsignaturen werden mittels Quantisierung zusammengefasst. Das Verfahren arbeitet auf Werten der Wavelet-Transformation, kann jedoch nach Aussage der Autoren auch auf die DCT und andere Transformationen angewandt werden.

Auf Basis eines geheimen Schlüssels wird die pseudo-zufällige Zahlensequenz B generiert. Seien p_i und q_i Koeffizienten unterschiedlicher Blöcke an den gleichen Positionen in der Zick-Zack-Anordnung und $B_i \in B$. Die Signatur wird dann wie folgt generiert:

$$Sig_i = \begin{cases} 0, & (p_i - q_i + B_i \geq 0) \\ 1, & (p_i - q_i + B_i < 0) \end{cases} \tag{3.13}$$

Im Verifikationsprozess muss das zu überprüfende Koeffizientenpaar (p'_i, q'_i) eine der folgenden Bedingungen erfüllen:

$$\begin{cases} (p'_i - q'_i + B_i > M) \wedge Sig_i = 0 & (Bedingung\ 1) \\ (|p'_i - q'_i + B_i| \leq M) & (Bedingung\ 2) \\ (p'_i - q'_i + B_i < -M) \wedge Sig_i = 1 & (Bedingung\ 3) \end{cases} \tag{3.14}$$

M repräsentiert in Gleichung (3.14) einen Schwellwert, um einen falschen Alarm durch Rauschen zu verhindern. $B_i \in B$ ist identisch mit den B_i aus dem Signatur-Generierungsprozess. Sollte keine der drei Bedigungen erfüllt werden, so wird das Koeffizientenpaar als manipuliert angesehen.

In der Erweiterung ihres Verfahrens führen die Autoren eine neue Quantisierungsmethode ein. Gleichung (3.13) wird folgendermaßen modifiziert:

$$Sig_i = \begin{cases} 0, & (p_i - q_i + B_i > Q) \\ 1, & (|p_i - q_i + B_i| \leq Q) \\ 2, & (p_i - q_i + B_i < -Q) \end{cases} \tag{3.15}$$

Q repräsentiert in Gleichung (3.15) einen Schwellwert, der die minimale Differenz zweier Koeffizienten p_i und q_i unter Berücksichtigung von B_i darstellt. Zusätzlich wird die dadurch entstandende Signatur mittels einer Tabelle in eine binäre Signatur überführt um anschließend mit dem in [LC00] beschriebenen Prozess als Wasserzeichen eingebettet zu werden.

Das veränderte Verfahren erhöht die Sensitivität gegenüber unerlaubten Veränderungen bei verbesserter Robustheit gegenüber erlaubten Veränderungen, wie z.B. JPEG2000-Kompression.

Evaluierung von Ekici et al.

Ekici et al. geben in [ESC+04] einen Überblick über bestehende semifragile Bildwasserzeichen. Sie stellen 8 Verfahren vor und vergleichen sie hinsichtlich ihrer Robustheit und Sensitivität. Sie konzentrieren sich dabei ausschließlich auf semifragile Verfahren, die statistisch verifizierbar sind. Das bedeutet, dass die Verfahren im Gegensatz zu visuell verifizierbaren Verfahren nicht mit Hilfe eines Logos oder Musters (wie in [YM97] und [Won98]) arbeiten. Neben dem Verfahren von Lin und Chang werden dort folgende weitere Verfahren untersucht:

- Lin, Podilchuk, Delp [LPD00]: pseudo-zufällige Rauschsequenz
- Eggers, Girod [EG01]: zufällige binäre Sequenz
- Fridrich, Goljan [FG00]: robuster visueller Hash
- Kundur, Hatzinakos [KH99]: zufällige binäre Sequenz

- Queluz [Que01]: zufällige binäre Sequenz
- Lan, Mansour, Tefik [LMT00]: auf Merkmalen basierende binäre Sequenz
- Yu, Lu, Liao [YLL01]: zufällige binäre Sequenz

Um die Verfahren hinsichtlich ihrer Robustheit gegenüber erlaubten Veränderungen zu evaluieren, wurden folgende Manipulationen auf den markierten Bildern durchgeführt:

- Weichzeichnungsfilter
- Schärfefilter
- 1% Salt-and-Pepper-Rauschen
- Histogramm-Ausgleich
- 35 dB weißes Rauschen
- JPEG-Kompression mit Qualitätsfaktor 70
- 0.1% der Bildbits wurden verändert

Um unerlaubte Veränderungen zu testen, führen Ekici et al. die Ersetzung eines Teils des markierten Bildes durch sein unmarkiertes Original durch. Dabei orientieren sich die Autoren an den Blockgrößen des jeweiligen Algorithmus, d.h. der Algorithmus von Lin und Chang wird mit einer Blockgröße von 8×8 Pixeln getestet, während mit dem Verfahren von Fridrich und Goljan ein Block der Größe 64×64 Pixeln ausgetauscht wird. Aus der Veröffentlichung wird jedoch nicht ersichtlich, welche Positionen und wieviele Blöcke pro Bild ausgetauscht wurden.

Pro Verfahren wurden 10 verschiedene Bilder mit 10 verschiedenen Schlüsseln markiert. Bei den erlaubten Veränderungen wird die Fehlerrate P_F (Probability of False Alarm) berechnet, die den Anteil der Bilder bezeichnet, die einen Fehler anzeigen, obwohl keine unerlaubte Veränderung durchgeführt wurde. Die Fehlerrate P_M (Probability of Miss) bezeichnet den Anteil der Bilder, die trotz unerlaubter Veränderung als authentisch verifiziert wurden. Dabei kommen Ekici et al. zu folgenden Ergebnissen:

- Lin, Chang: Gute Robustheit gegenüber JPEG-Kompression (0,1%) und weißem Rauschen (8,7%). Hohe Fragilität gegenüber Signalverarbeitungs-Operationen wie Histogramm-Ausgleich (69,4%) und Schärfefilter (89,1%). Gute Sensitivität bei unerlaubter Veränderung (11,3%).

- Lin, Podilchuk, Delp: Gute Robustheit gegenüber Signalverarbeitungs-Operationen, außer gegenüber Weichzeichnungsfilter (69,1%). Gute Sensitivität bei unerlaubter Veränderung (6,4%). Allerdings ist auch eine hohe Fehlerrate zu erkennen, wenn weder eine erlaubte, noch eine unerlaubte Veränderung auf das markierte Bild durchgeführt wurde (14,3%).
- Eggers, Girod: Gute Robustheit gegenüber JPEG-Kompression (0,2%) und weißem Rauschen (8,6%). Geringe Robustheit gegenüber Signalverarbeitungs-Operationen, wie Histogramm-Ausgleich (67,1%) und Schärfefilter (59,7%). Mäßige Sensitivität bei unerlaubter Veränderung (24,9%).
- Fridrich, Goljan: Gute Robustheit gegenüber weißem Rauschen (2,5%) und Histogramm-Ausgleich (5,5%). Sehr gute Sensitivität bei unerlaubter Veränderung (1,0%).
- Kundur, Hatzinakos: Mäßige Robustheit gegenüber erlaubten Veränderungen (z.B. JPEG-Kompression 14,9%, Histogramm-Ausgleich 59,0%) und auch mäßige Sensitivität bei unerlaubter Veränderung (37,3%).
- Queluz: Sehr gute Robustheit gegenüber erlaubten Veränderungen, außer gegenüber Histogramm-Ausgleich (41,6%) und Schärfefilter (50,3%). Schlechte Sensitivität bei unerlaubter Veränderung (49,9%).
- Lan, Mansour, Tefik: Mäßige Robustheit gegenüber erlaubten Veränderungen (z.B. weißes Rauschen 85,5%) bei guter Sensitivität (16,9%).
- Yu, Lu, Liao: Gute Robustheit gegenüber Signalverarbeitungs-Operationen, außer gegenüber Schärfefilter (62,7%) und Histogramm-Auslgeich (63,1%). Mäßige Sensitivität bei unerlaubter Veränderung (26,2%).

3.2.5 Fragile Wasserzeichen für Videos

Verfahren von Celik et al.

Aufbauend auf der Arbeit von Wong [Won98] und Fridrich [Fri02] stellen Celik et al. in [CSST02] ein fragiles Videowasserzeichen vor. Das auf Blöcken basierende Verfahren kann sowohl Manipulationen von Einzelbildern als auch Manipulationen an deren zeitlicher Abfolge erkennen.

Der Einbettungsprozess wird in zwei Schritte unterteilt: die Erstellung der Wasserzeichennachricht S, d.h. die Inhaltsbeschreibung, und deren Einbettung.

Sei L_{Packet} die Positionsinformation eines Blockes, die das Video, das Frame, die Blockposition, Videoauflösung und die Anzahl der Bilder im Video eindeutig beschreibt. Sei ferner R_{Packet} die Information zur Wiederherstellung des entsprechenden Blockes. Seien darüber hinaus MSB die höchsten Bitebenen, Key ein geheimer Schlüssel, $\mathcal{H}$ eine Hashfunktion, $\mathcal{E}$ eine Verschlüsselungsfunktion und $\oplus$ die XOR-Operation. Dann wird die Blocksignatur S wie folgt gebildet:

$$S = \mathcal{E}\left(\mathcal{H}\left(MSB \,\|R_{Packet}\right) \oplus L_{Packet}, Key\right) \tag{3.16}$$

S wird mittels eines Algorithmus in die niedrigste Bitebene (LSB) eines zugehörigen Blockes eingebettet. Die Position des zugehörigen Blockes wird wie in Gleichung (3.17) berechnet. T repräsentiert hier eine drei-dimensionale Matrix, die einen horizontalen, vertikalen und zeitlichen Abstand bestimmt. s ist ein geheimer Vektor, der das Positionsergebnis verschiebt und N repräsentiert den maximalen Abstand. Der Prozess wird als Interleaving bezeichnet.

$$y = Tx + s \;\; (mod\ N) \tag{3.17}$$

Die Wiederherstellungsinformation R_{Packet} wird durch ein so genanntes Multiple-Description-Coding von Keyframes realisiert. Keyframes repräsentieren den Inhalt von Videosequenzen und werden beispielsweise bei der Video-Segmentierung und -Indexierung verwendet [FTM98]. Die Information wird mittels Multiple-Description-Coding zerteilt um bei Verlust einer Teilinformation immer noch die Möglichkeit zu haben die Restinformation wiederherzustellen. Celik et al. geben ein einfaches Beispiel in Form eines 16×16 Blockes, dessen Koeffizienten nach JPEG komprimiert werden. Dabei werden jeweils die geraden und ungeraden AC-Koeffizienten mit dem DC-Koeffizienten zu Teilinformationen zusammengefasst. Gehen beispielsweise die geraden AC-Koeffizienten verloren so können zumindest der DC-Koeffizient und die ungeraden AC-Koeffizienten wiederhergestellt werden. Aus [CSST02] geht jedoch nicht hervor, wie die Position der zugehörigen Blöcke bestimmt wird, wenn mehr als ein Paket mit Wiederherstellungsinformationen vorhanden ist.

Im Verifikationsprozess wird die Positionsinformation L_{Packet} wie folgt bestimmt, wobei $\mathcal{D}$ die zu $\mathcal{E}$ gehörende Entschlüsselungsfunktion repräsentiert:

$$L_{Packet} = \mathcal{D}\left(S, Key\right) \oplus \mathcal{H}\left(MSB \,\|R_{Packet}\right) \tag{3.18}$$

Das Verfahren kann sehr gut Veränderungen am Videomaterial erkennen. Sowohl räumliche als auch zeitliche Veränderungen werden erkannt und der Original-Inhalt kann in einer Version mit geringerer Qualität wiederhergestellt werden. Das Verfahren ist sowohl auf unkomprimiertes als auch auf komprimiertes Videomaterial (bspw. MPEG-Videos) anwendbar. Durch seine Fragilität wird jedoch das Wasserzeichen durch Formatumwandlung oder Reenkodierung zerstört.

Verfahren von Mobasseri und Evans

In [ME01] stellen Mobasseri und Evans ein fragiles Videowasserzeichen vor, das Framepaare (f, f^*) bildet und eine reduzierte Version von f in den Farbraum von

f^* als Wasserzeichen einbettet. Ziel des Verfahrens ist es so genannte Re-Indexing-Angriffe zu erkennen, also Angriffe, die Frames entfernen, einfügen oder vertauschen. Bei einem erfolgten Angriff soll die reduzierte Version wiederhergestellt werden um den Original-Inhalt zu erkennen.

Sei f ein Frame eines Videos mit 24 Bit Farbtiefe und f^* das zum Framepaar (f, f^*) zugehörige Frame. Aus f wird durch Re-Quantisierung ein Frame f_q mit 3 Bit Farbtiefe generiert. Sei $c = [c_1, c_2, c_3]$ eine Matrix, deren Elemente aus 2D M-Sequenzen [Gol82] besteht. Dann wird das Wasserzeichen f_w, das den Inhalt von f beschreibt, wie folgt berechnet:

$$f_w = f_q \cdot c = \left[f_{q1} \cdot c_1, f_{q2} \cdot c_2, f_{q3} \cdot c_3 \right] \tag{3.19}$$

Kontrolliert durch einen geheimen Schlüssel wird f_w in eine der unteren drei Bitebenen des Rot-, Grün- und Blau-Kanals eingebettet, indem die Original-Bitwerte durch f_w ersetzt werden. Das Wasserzeichen wird somit über die Farbkanäle von f^* gespreizt. Die Verifikation erfolgt auf gleiche Weise. Aus Frame f wird eine reduzierte Version f_q mittels Re-Quantisierung erzeugt und mit der Spreizmatrix c multipliziert. f_w wird aus f^* ausgelesen und mit dem Matrizenprodukt $f_q \cdot c$ verglichen. Stimmen sie nicht überein, so liegt eine Manipulation vor.

Die Frame-Paare (f, f^*) werden durch Unterteilung des Videos in Segmente der Länge N und einen geheimen Schlüssel zugeordnet. Dabei hat jedes Frame f ein zugehörendes Frame f^* im nachfolgenden Segment. Sollte ein Frame durch einen Re-Indexing-Angriff nicht mehr vorhanden sein oder durch andere Veränderungen beschädigt worden sein, so wenden die Autoren verschiedene Suchstrategien an, um das korrespondierende Frame zu finden, das den reduzierten Inhalt als Wasserzeichen enthält.

Das Verfahren wurde nur für den Schutz von unkomprimierten Videos entwickelt. Zwar testeten die Autoren das Wasserzeichen auch in komprimierten Videos, indem sie das Wasserzeichen auf alle drei Bitebenen der drei Farbkanäle verteilten und nicht nur auf eine der drei Bitebenen. Allerdings war das Video intrakodiert, bestand also nur aus I-Frames, und wurde bereits nach JPEG-Kompression mit einem Qualitätsfaktor von 50% sichtbar.

3.2.6 Semifragile Wasserzeichen für Videos

Verfahren von Lin und Chang

In [LC99] stellen Lin und Chang ein semifragiles Signatur-Verfahren für MPEG-Videos vor, das auch als Ausgangsbasis für ein semifragiles Videowasserzeichen dienen kann. Zunächste definieren die Autoren 5 erlaubte Manipulationen an MPEG-Videodaten:

1. Die höchsten DCT-Koeffizienten in 8×8 Blöcken werden entfernt, um die Bitrate eines MPEG-Videos dynamisch anzupassen. Die Bewegungs-Vektoren werden nicht modifiziert.
2. Die DCT-Koeffizienten werden re-quantisiert, um die Bitrate zu verändern. Hierbei werden ebenfalls die Bewegungs-Vektoren nicht modifiziert.
3. Ähnlich zum vorhergehenden Szenario werden die DCT-Koeffizienten re-quantisiert. Allerdings werden die inter-codierten Blöcke derart modifiziert, dass die Videoqualität nach der Re-Quantisierung gleich bleibt. Auch hier werden die Bewegungs-Vektoren nicht modifiziert.
4. In diesem Szenario werden Schnitte an Sequenzen durchgeführt, wobei die ursprünglichen Bildtypen (I-, P- und B-Frames) erhalten bleiben. Die Group of Pictures (GOP) bleiben nahezu erhalten. Ausnahmen bilden die GOPs, die von den Schnitten betroffen sind. Die Pixelwerte können verändert werden, um die Videoqualität zu verbessern.
5. Ähnlich zum vorhergehenden Szenario werden Schnitte an Sequenzen durchgeführt und eventuell Pixelwerte verändert. Allerdings werden hier die ursprünglichen Bildtypen nicht beibehalten. Für dieses Szenario stellen die Autoren eine abgewandelte Version der digitalen Signatur vor.

Die Erstellung der Signatur beruht auf der Erkenntnis der Autoren in [LC98], dass DCT-Koeffizienten an der gleichen Position verschiedener Blöcke das selbe Verhältnis nach JPEG-Kompression aufweisen (siehe dazu auch Abschnitt 3.2.4). Dies gelte ebenso für DCT-Koeffizienten in MPEG-Videos. Das Verfahren arbeitet auf den quantisierten DCT-Koeffizienten von 8×8 Blöcken. Sei $f_p(b)$ der DCT-Koeffizient an Position b in Block p und $f_{W(p)}(b)$ der DCT-Koeffizient an Position b im durch die Funktion $W(p)$ zugeordneten Block. Dann wird das Merkmal Z_c eines Frames wie folgt berechnet:

$$Z_c = VLC\left(\bigcup_p \bigcup_b sgn\left[f_p(b) - f_{W(p)}(b)\right]\right) \tag{3.20}$$

VLC ist als variable Längencodierung definiert, um das Ergebnis zu komprimieren. Die Funktion sgn ist wie folgt definiert:

$$sgn(f) = \begin{cases} 1 & \\ 0 & if \\ -1 & \end{cases} \begin{array}{l} f_p(b) - f_{W(p)}(b) > 0 \\ f_p(b) - f_{W(p)}(b) = 0 \\ f_p(b) - f_{W(p)}(b) < 0 \end{array} \tag{3.21}$$

Zusätzlich zu den Verhältnissen der DCT-Koeffizienten werden aus den zusätzlichen Bildinformationen, wie Bewegungsvektoren und Frameposition ein Hashwert Z_m berechnet. Die Signatur DS ist das Ergebnis der Konkatenation von Z_c und Z_m nach einer Verschlüsselung mit einem privaten Schlüssel.

Für Szenario 5 wird das Signaturverfahren dahingehend abgewandelt, dass die Pixelwerte aus den I-, P- und B-Frames wiederhergestellt werden und erst dann das Merkmal Z_c nach erfolgter DCT berechnet wird. Z_m enthält hierbei die Frameposition.

Das vorgestellte Signaturverfahren kann zwischen Veränderungen durch MPEG-Kompression und Veränderungen des Inhalts unterscheiden. Die Signatur kann als Merkmal für ein semifragiles Wasserzeichenverfahren verwendet werden, sofern das Wasserzeichen den Bereich nicht verändert, der zur Berechnung der Signatur verwendet wird. Allerdings ist sie sehr formatabhängig, da sie speziell für MPEG-Videos entwickelt wurde.

Verfahren von Dittmann et al.

In [DSS99] stellen Dittmann et al. ein auf extrahierten Kanten basierendes semifragiles Videowasserzeichen vor.

Der Kantendetektor von Canny [Can86] liefert für jedes Pixel eines Eingabebildes oder Eingabeframes I einen Grauwert zurück der besagt, ob das Pixel auf einer Kante (grau oder schwarz) oder im Hintergrund liegt (weiß). Das Grauwertbild wird als Kantencharakteristik C_I bezeichnet. Zur weiteren Verarbeitung überführen die Autoren C_I in eine binäre Abbildung EP_{CI}. Die binäre Abbildung wird mittels einer variablen Längencodierung VLC komprimiert. Das Resultat der Längencodierung $VLC(EP_{CI})$ dient als Merkmal für das Wasserzeichen.

Für die Einbettung des Merkmals und zusätzlichen Informationen über den Rechteinhaber verwenden die Autoren sich überlagernde Muster der Größe 8×8, wie in [DSS98] beschrieben. Die Generierung der Muster ähnelt dem Verfahren von Fridrich aus [Fri97]. Es werden die Muster $M_{Content}$ und $M_{Private}$ generiert: $M_{Content}$ mit einem geheimen Schlüssel $PrivateKey$ und dem Merkmal $VLC(EP_{CI})$ und $M_{Private}$ mit $PrivateKey$ und den Informationen über den Rechteinhaber. Das markierte Videoframe I_W wird wie folgt gebildet:

$$I_W = I + M_{Content} + M_{Private} \tag{3.22}$$

Im Verifikationsprozess wird $M_{Content}$ mit dem Wasserzeichenalgorithmus detektiert. Zusätzlich wird die aktuelle Kantencharakteristik berechnet, komprimiert und daraus ein Muster $M'_{Content}$ generiert. Stimmen die beiden Muster $M_{Content}$ und $M'_{Content}$ überein, so kann davon ausgegangen werden, dass der Inhalt des Frames nicht verändert wurde.

Das vorgestellte Verfahren bietet nicht die Möglichkeit Veränderungen am Inhalt zu lokalisieren. Darüber hinaus besteht als offenes Problem die mangelnde Resistenz gegenüber Skalierung und starker Kompression, da sich durch beide inhaltserhaltende Operationen die Kanten leicht verschieben können. Gleichzeitig werden

Veränderungen an Farben nicht erkannt, da sich das Merkmal nur auf Kanten bezieht. Dadurch können Veränderungen am Inhalt (z.B. falsche Flagge bei einer Nachrichtensendung) nicht erkannt werden. Darüber hinaus werden keine Strategien aufgezeigt, wie Veränderungen an der zeitlichen Abfolge von Frames erkannt werden können.

In [DSR+00] wird das Verfahren um ein inhaltsbeschreibendes Merkmal des Audiokanals erweitert. Es wird ermittelt, an welchen Stellen des Audiofrequenz-Signals das Vorzeichen wechselt, das so genannte Zero-Crossing. Diese Positionen werden zu einem Merkmalsvektor zusammengefasst. Der wie in [DSS99] generierte Merkmalsvektor des Videoinhalts und der Merkmalsvektor des Audioinhalts werden mittels einer XOR-Operation verknüpft und in beide Kanäle mittels robuster Wasserzeichenverfahren eingebettet. Dittmann et al. beschreiben dazu in [DSR+00] einen Algorithmus zum Markieren von Audiodaten, codiert nach MPEG Audio Layer 2. Zusätzlich schlagen die Autoren in [DSR+00] vor, einen 16 Bit langen Zeitstempel in jedes Frame einzubetten um Manipulationen an der zeitlichen Abfolge sowohl im Audio- als auch im Videokanal zu erkennen.

Verfahren von Yin und Yu

In [YY02] stellen Yin und Yu ein semifragiles Videowasserzeichen vor, das speziell für MPEG-Videoströme konzipiert wurde und ein fragiles und ein robustes Wasserzeichen kombiniert. Die Grundidee liegt darin, dass zunächst das fragile Wasserzeichen Auskunft darüber gibt, ob eine Manipulation stattgefunden hat und bei angezeigter Manipulation deren Position mit dem robusten Wasserzeichen lokalisiert wird. Zu beachten ist hier, dass es sich nicht um ein semifragiles Wasserzeichen in unserem Sinne handelt, d.h. um ein semifragiles Merkmal das mit einem robusten Wasserzeichen eingebettet wird. Mit ihrem Konzept möchten die Autoren insbesondere Umwandlungsprozesse, d.h. MPEG-Umwandlung mit gleich bleibenden Frametypen, von anderen Manipulationen unterscheiden können.

Die Autoren führen drei Arten von Merkmalen ein:

1. Blockmerkmal: Das Blockmerkmal enthält Informationen über den Inhalt von I-Frames auf Blockebene. Dadurch ist eine Lokalisierung von Manipulationen an den Blöcken möglich. Das Merkmal, das die Autoren in ihren Tests verwendeten, wird nicht näher beschrieben.

2. Framemerkmal: Das Framemerkmal enthält Informationen über den Inhalt auf Frameebene. Dazu wird aus den höchsten Bitebenen (MSB) der AC-Koeffizienten eines Frames ein Hashwert mittels einer Hashfunktion berechnet. Das Framemerkmal wird ebenfalls nur für I-Frames berechnet und besteht aus den verschlüsselten Informationen der Block- und der Frameebene.

3. GOP-Merkmal: Für das GOP-Merkmal werden auch die inhaltsbeschreibenden Framemerkmale für die P- und B-Frames berechnet. Aus den Merkmalen

aller Frames einer GOP wird mittels einer Hash-Funktion ein Hashwert berechnet. Zusätzlich werden die ersten m Bits der einzelnen Hashwerte I-, P- und B-Frames verwendet und mittels Konkatenation zusammengefasst. Der dritte Teil des GOP-Merkmals besteht aus zusätzlichen Informationen über die GOP, wie beispielsweise zeitliche Informationen, Nummer des Videos oder Anzahl der Frames in der GOP. Alle drei Teilinformationen werden wiederum verschlüsselt.

Alle drei Merkmale werden von den Autoren zum fragilen Merkmal M_F zusammengefasst. Das Framemerkmal des I-Frames I_{i+1} aus GOP_{i+1} wird in I_{i+1} mittels eines LSB-Wasserzeichens eingebettet. Zusätzlich wird das GOP-Merkmal aus GOP_i eingebettet.

Für das robuste Merkmal M_R werden aus den, mittels eines geheimen Schlüssels vermischten, DC-Koeffizienten des I-Frames I_{i+1} ein Hashwert berechnet. Der verschlüsselte Hashwert wird mittels eines robusten Wasserzeichenverfahrens eingebettet.

Bei der Wahl von M_F und M_R und deren Wasserzeichenverfahren ist darauf zu achten, dass sich die jeweiligen Verfahren und Merkmale nicht gegenseitig beeinflussen. So sollte beispielsweise der Einbettungsprozess von M_R nicht das Ausleseergebnis von M_F verändern. Das Verfahren beschränkt sich ausschließlich auf MPEG-Videos deren Frametypen nach einer Transcodierung gleich bleiben. Eine Konvertierung in andere Videoformate oder eine Veränderung der Frametypen wird als unerlaubte Manipulation angesehen werden. Interessant jedoch ist die Kombination von robusten und fragilen Merkmalen und von robusten und fragilen Wasserzeichenverfahren.

Verfahren von Huang et al.

In [HHP08] stellen Huang et al. ein semifragiles Videowasserzeichen vor, das für Videos vom Typ Motion-JPEG entwickelt wurde. Bei diesem Kompressionsverfahren wird jedes Frame eines gegebenen Videos mittels JPEG komprimiert. Obwohl Motion-JPEG über eine geringere Kompressionsrate als bspw. MPEG verfügt wird es sehr häufig in Überwachungssystemen eingesetzt, da es auf ein Frame genau geschnitten werden kann und eine gleichbleibende Qualität unabhängig von Bewegungen liefert.

Huang et al. betten in jedes Frame F ein robustes Videowasserzeichen W_f ein, das sich aus der Framenummer S innerhalb einer Videosequenz und der Anzahl der Frames in der Sequenz T zusammensetzt. Sowohl S als auch T haben eine Länge von 20 Bit.

Es wird vorausgesetzt, dass jedes Videoframe im RGB-Format vorliegt. In jedem 8×8 DCT-Block im Rot-, Grün- und Blaukanal werden zwei DCT-Koeffizienten modifiziert um ein Wasserzeichenbit $W_b \in W_f$ einzubetten. Sei X ein DCT-Koeffizient,

X' der zugehörige markierte DCT-Koeffizient und λ ein Schwellwert. Dann wird W_b wie folgt eingebettet:

$$X' = \begin{cases} X & ((W_b = 1 \wedge X \geq \lambda) \vee (W_b = 0 \wedge X \leq -\lambda)) \\ \lambda & if \quad (W_b = 1 \wedge X < \lambda) \\ -\lambda & (W_b = 0 \wedge X > -\lambda) \end{cases} \tag{3.23}$$

Die Zuordnung eines Bits W_b zu einem 8×8 DCT-Block wird durch einen geheimen Schlüssel K_1 kontrolliert. Die Auswahl der zu modifizierenden DCT-Koeffizienten wird durch einen zweiten geheimen Schlüssel K_2 kontrolliert. Jedes Bit wird an der gleichen Position im Rot-, Grün- und Blaukanal eingebettet.

Im Ausleseprozess wird aus jedem markierten Koeffizienten das Wasserzeichenbit W_b' ausgelesen:

$$W_b' = \begin{cases} 1 & (X' > \lambda_1) \\ -1 & if \quad (X' < -\lambda_1) \\ 0 & (|X'| < \lambda_1) \end{cases} \tag{3.24}$$

Den Schwellwert λ_1 geben die Autoren mit $\lambda_1 = \lambda/2$ an. Ob ein Block manipuliert wurde, kann mit folgender Gleichung bestimmt werden:

$$W_S' = \begin{cases} 1 & (\sum W_b' \geq \lambda_2) \\ -1 & if \quad (\sum W_b' \leq -\lambda_2) \\ 0 & (|\sum W_b'| < \lambda_2) \end{cases} \tag{3.25}$$

λ_2 ist ein Schwellwert, der bestimmt, ob ein Block manipuliert wurde. Er wird von den Autoren mit $\lambda_2 = 4$ angegeben. Wie aus Gleichung (3.25) ersichtlich wird, bestimmt das Verfahren aufgrund einer Mehrheitsentscheidung, ob der Block in den drei Farbkanälen manipuliert wurde, nämlich wenn der Betrag der Summe aller W_b' den Schwellwert λ_2 unterschreitet. Das Bit des semifragilen Wasserzeichens W_S' wird schließlich verwendet, um aufgrund einer Mehrheitsentscheidung S und T zu bestimmen. Hiermit können Manipulationen an der Zeitachse lokalisiert werden.

Das Verfahren zeigt eine gute Robustheit gegenüber JPEG-Kompression bis zu einem JPEG-Qualitätsfaktor von 55 und gegenüber additivem Rauschen. Da T erst nach Abschluss einer Aufnahme feststehen kann, ist der Einsatz des Verfahrens in Echtzeit-Überwachungsszenarien zu hinterfragen. Unklar ist darüber hinaus, warum die Autoren auf dem RGB-Farbraum arbeiten obwohl JPEG bzw. Motion-JPEG im YC_BC_R-Farbraum arbeiten. Zudem werden zur Generierung des Wasserzeichens keine Informationen über den Inhalt verwendet sondern strukturelle Informationen des Videos.

3.2.7 Weitere Verfahren

In [KTB07] verwenden Kitanovski et al. die DC-Koeffizienten von 8×8 DCT-Blöcken um ein semifragiles Merkmal zu extrahieren. Aus dem Verhältnis zwischen DC-Koeffizienten benachbarter Blöcke wird ein binärer Vektor gebildet, der als Schlüssel für die Generierung eines Wasserzeichens dient. Verändert sich der Merkmalsvektor geringfügig, so verändert sich das komplette Wasserzeichen. Daher ist eine auf einen Block genaue Lokalisierung einer Manipulation nicht möglich. Das Wasserzeichen wird mittels Quantization Index Modulation (QIM) [CW01] eingebettet.

Pröfrock et al. stellen in [PRSM05] ein fragiles Videowasserzeichen vor, das speziell für Videos vom Typ H.264/AVC entwickelt wurde. Dazu wird aus den Videodaten ein Hashwert gebildet, der dann mittels Public-Key-Encryption verschlüsselt wird. Der verschlüsselte Hashwert wird mittels QIM in Macroblöcke eingebettet, die keine Daten aufgrund der Bewegungskompensation enthalten und dadurch bei der Dekodierung des Videos übersprungen werden.

Ein weiteres semifragiles Videowasserzeichen wird von Mobasseri et al. in [MSS00] vorgestellt. Als Wasserzeichen dient hier eine 2D-Bitmap die mit Hilfe einer Spread-Spectrum-Technik auf verschiedene Frames verteilt wird. Das Verfahren dient zum Erkennen von ausgeschnittenen bzw. eingefügten Framesequenzen.

3.3 Zusammenfassung

Die in diesem Kapitel vorgestellten Integritätswasserzeichen weisen zum Teil eine gute Sensitivität mit einer möglichen Lokalisierbarkeit auf. Allerdings sind diese Verfahren entweder komplett fragil oder sehr abhängig vom zugrundeliegenden Format. Ziel dieser Arbeit ist es daher ein format-unabhängiges Verfahren zu entwickeln, das den Videoinhalt beschreibt und damit schützt und zugleich die Zeitachse schützt. Das Verfahren soll eine gute Robustheit aufweisen bei gleichzeitiger Möglichkeit die manipulierten Stellen zu lokalisieren.

Kapitel 4

Entwurf

Analog zu Steinebach [Ste04] definieren wir den Begriff Integrität einer Videodatei wie folgt:

Solange der Inhalt einer Videodatei nicht verändert worden ist, ist die Integrität vorhanden.

Dabei beziehen wir uns auf die Definition des Inhalts aus Kapitel 1.1, d.h. also dass der Inhalt die Kernaussage der sichtbaren Informationen ist. Die Kernaussage ist abhängig vom zugrundeliegenden Szenario. So sollten nach unserer Auffassung alle Objekte historischer Aufnahmen geschützt werden. Im Falle von Überwachungsaufnahmen können Objekte im Vordergrund bzw. die Uhrzeit geschützt werden. Im Idealfall sollte der Urheber bestimmen, was die Kernaussage eines Videos ist.

In diesem Kapitel stellen wir den Entwurf unseres Integritätswasserzeichenverfahrens vor und diskutieren Anforderungen an das Wasserzeichenverfahren und das inhaltsbeschreibende Merkmal.

4.1 Konzeptentwurf

Ausgehend von den in [DS00] von Dittmann und Steinebach vorgestellten Verfahren für Bilder entwerfen wir in diesem Abschnitt ein Konzept für ein digitales Wasserzeichen zum Schutz der Integrität digitaler Videos. Dittman et al. generieren auf Basis von verschiedenen Charakteristiken (z.B. Anzahl von Kanten in einzelnen Blöcken oder Varianz der Kantenintensität im gesamten Bild) den Merkmalsvektor für ein zu schützendes Bild. Dieser wird mittels eines robusten Wasserzeichenverfahrens in das zu schützende Bild eingebettet. Wir übertragen in dieser Arbeit das Konzept auf Videos.

Abbildung 4.1 stellt schematisch den Entwurf für den Einbettungsprozess dar. Sei V das zu schützende Video. Seien darüber hinaus K_1 und K_2 geheime Schlüssel,

die nicht notwendigerweise unterschiedlich sein müssen. Aus Video V wird mittels eines Generierungsalgorithmus G, der durch den Schlüssel K_1 kontrolliert wird, der Merkmalsvektor M generiert. Der Generierungsalgorithmus G ist parametriesierbar, d.h. zu authentifizierende Bildregionen können festgelegt werden. Im folgenden beschränken wir uns auf Algorithmen, die zunächst die gesamten Frames authentifizieren. M repräsentiert in binärer Form Informationen über den Inhalt von V. Mittels eines Einbettungsalgorithmus E, der durch den Schlüssel K_2 kontrolliert wird, wird der Merkmalsvektor M als robustes Wasserzeichen in das Video V eingebettet. Das Ergebnis von E ist das markierte Video V_W.

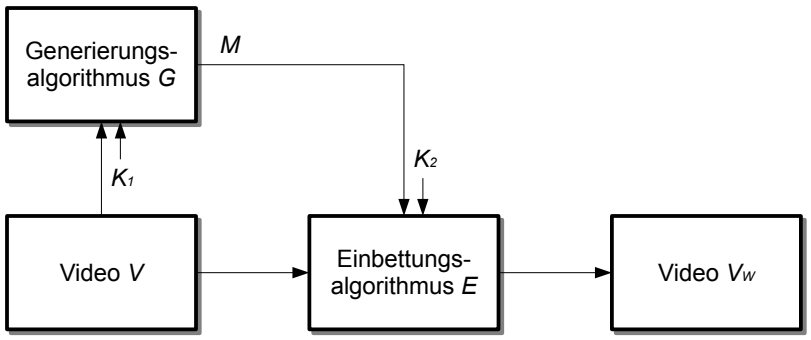


Abbildung 4.1: Entwurf für den Einbettungsprozess

Abbildung 4.2 stellt schematisch den Entwurf für den Detektionsprozess dar. Wir übernehmen aus dem Einbettungsprozess die geheimen Schlüssel K_1 und K_2 sowie den Generierungsalgorithmus G. Sei V'_W das markierte und möglicherweise manipulierte Video. Aus V'_W wird mittels Generierungsalgorithmus G und dem ihn kontrollierenden Schlüssel K_1 der Merkmalsvektor M' generiert. Mittels eines Detektionssalgorithmus D, der durch den Schlüssel K_2 kontrolliert wird, wird der Merkmalsvektor M aus dem Video V'_W ausgelesen. Anschließend entscheidet der Vergleichsalgorithmus C anhand von M und M' ob V'_W manipuliert wurde und wenn ja, an welchen Positionen.

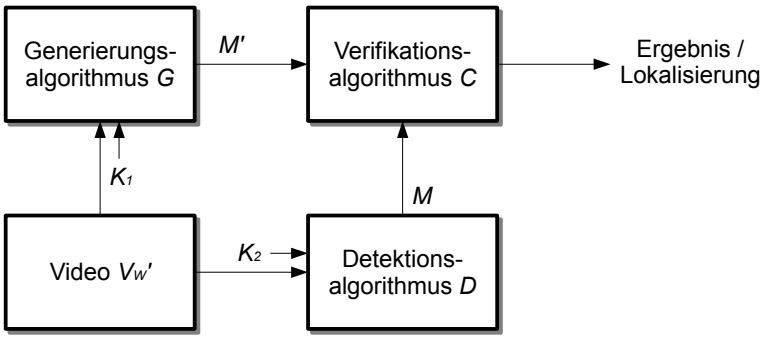


Abbildung 4.2: Entwurf für den Detektionssprozess

4.2 Anforderungen an das robuste Videowasserzeichen

Das ideale robuste Wasserzeichenverfahren, das den Merkmalsvektor M in das Video V einbettet bzw. aus dem Video V'_W ausliest sollte über verschiedene Eigenschaften verfügen.

Das Verfahren sollte robust gegenüber Veränderungen sein, die die Repräsentation des Videos verändern, jedoch dessen Inhalt beibehalten. Dazu zählen:

- **Formatkonvertierung**, z.B. Konvertierung von MPEG-2 in H.264 aber auch Digital/Analog - Analog/Digital-Wandlung
- **Veränderung der Bit- und Framerate** durch verlustbehaftete Kompression oder Formatänderung (z.B. von PAL nach NTSC)
- **Bildrauschen und Pixelfehler**, z.B. durch Übertragungsfehler oder verlustbehaftete Kompression

Eine zweite Gruppe von Veränderungen, die den Inhalt beibehalten sind Post-Production-Operationen. Dazu zählen:

- **Weichzeichenfilter**, z.B. zur Rauschunterdrückung
- **Anpassung der Gesamt-Helligkeit und des Gesamt-Kontrastes**, z.B. zur besseren Sichtbarkeit. Diese Veränderungen können abhängig vom Kontext jedoch wieder als inhalts-verändernd eingeordnet werden, d.h. wenn sie zu stark sind und damit die Aussage des Videos verändern (siehe dazu Abschnitt 4.3).

Die dritte Gruppe von Veränderungen behalten den Inhalt nicht bei. Trotzdem sollte das ideale robuste Wasserzeichenverfahren den Merkmalsvektor auch nach diesen Veränderungen auslesen können, um die Positionen der Veränderungen zu lokalisieren:

- **Drehung**
- **Skalierung**
- **Ausschnittbildung (Cropping)**
- **Entfernen von Frames, Szenen und Objekten aus dem Video**

Neben diesen starken Robustheitsanforderungen sollte das ideale Wasserzeichenverfahren zusätzlich eine möglichst hohe Kapazität aufweisen, um den Merkmalsvektor einbetten zu können. Das Wasserzeichen sollte nicht sichtbar sein. In Kapitel 5 beschäftigen wir uns mit der Entwicklung von robusten Wasserzeichenverfahren für Videos und analysieren später deren Eigenschaften im Vergleich zu den hier aufgestellten Anforderungen.

4.3 Anforderungen an das inhaltsfragile Merkmal

Das ideale inhaltsfragile Merkmal sollte Veränderungen, die den Inhalt beibehalten, auch als solche erkennen, d.h. die Integrität der Videodatei als vorhanden anzeigen. Dies sind die ersten beiden Gruppen von Veränderungen, die im vorherigen Abschnitt besprochen wurden.

Zusätzlich sollte das ideale inhaltsfragile Merkmal zwei Gruppen von Manipulationen erkennen, die den Videoinhalt verändern. Die erste Gruppe sind Manipulationen, die auf einzelne Frames ausgeführt werden und auch im Bildbereich Anwendung finden:

- **Manipulation an Objekten**, z.B. Zeitstempel bei Überwachungsaufnahmen
- **Veränderung der Helligkeit** - kann die Stimmung eines Videos ändern
- **Farbänderungen**, z.B. Einfärben von Flächen oder Veränderung von Flaggenfarben

Ein Beispiel für eine Manipulation an Objekten ist in Abbildung 4.3 zu sehen. Hier wurde auf dem rechten Bild die Uhrzeit geändert. In einer Videosequenz ist durch den Einsatz von Maskentechniken solch eine Manipulation sehr schnell zu realisieren.

Die Auswirkung von Farbänderungen auf den Betrachter wurde im Nachgang eines blutigen Anschlages auf Touristen in Luxor (Ägypten), am 17.11.1997 deutlich. Die Schweizer Zeitung „Blick“ veröffentlichte Fotos mit veränderten Farben. Eine Wasserlache wurde hierbei rötlich eingefärbt. In der Beschreibung des Bildes wurde die rötlich eingefärbte Wasserlache als Blutsspur des Massakers dargestellt[2].

Die zweite Gruppe sind Manipulationen, die sich speziell auf Videos beziehen. Sie manipulieren die Zeitachse. Dazu zählen:

- **Manipulationen am Ablauf der Framewiedergabe**, z.B. durch das Vertauschen von Szenen

[2]http://www.rhetorik.ch/Bildmanipulation/Bildmanipulation.html (Aufruf am 02.07.2011)

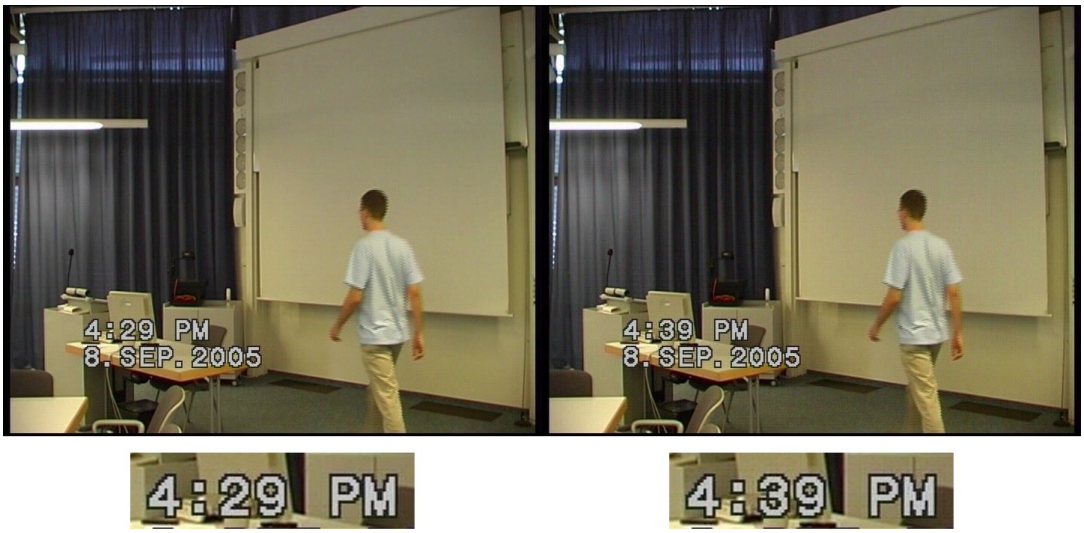


Abbildung 4.3: Beispiel für eine Objektmanipulation

- **Entfernen oder Hinzufügen einzelner Frames oder Szenen**, z.B. im Kontext von Nachrichtensendungen (siehe Kapitel 2.2)

In Kapitel 6 beschäftigen wir uns mit der Entwicklung inhaltsfragiler Merkmale. Später analysieren wir die Eigenschaften der entwickelten Verfahren im Vergleich zu den in diesem Abschnitt aufgestellten Anforderungen.

Kapitel 5

Robuste Videowasserzeichen

Im Rahmen dieser Arbeit wurden zwei robuste Videowasserzeichenverfahren entwickelt. In Abschnitt 5.1 beschäftigen wir uns mit der Verbesserung des additiven Verfahrens von Fridrich. Dieses wird später für die Einbettung der Merkmalsvektoren verwendet. In Abschnitt 5.2 stellen wir ein weiteres Verfahren vor, das auf Basis von DCT-Koeffizienten-Relationen arbeitet.

5.1 Verbesserung des additiven Verfahrens von Fridrich

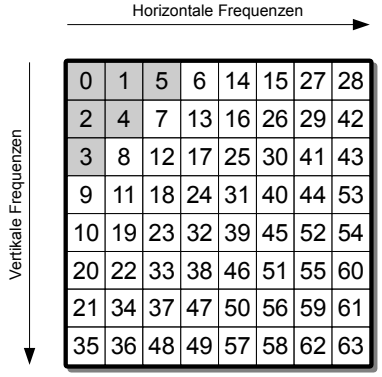
In Kapitel 3.1.1 beschäftigten wir uns mit dem robusten Wasserzeichenverfahren von Fridrich [Fri97], welches ein nicht wahrnehmbares Rauschmuster auf ein Bild aufträgt. Dieses Verfahren wurde von Dittmann [Dit00] auf Videos erweitert. Zusätzlich benötigte die Erweiterung von Dittmann nicht mehr das Original im Ausleseprozess.

In diesem Abschnitt stellen wir Verbesserungen des Verfahrens in Bezug auf die Verfahrensparameter Transparenz, Robustheit und Kapazität vor. Das Ziel besteht darin das Verfahren für den Einsatz als robustes Videowasserzeichen in unserem in Kapitel 4 vorgestellten Konzept zu optimieren.

5.1.1 Verbesserung der Transparenz

Die Diskrete Kosinustransformation (Discrete Cosine Transformation / DCT) überführt die Bilddaten eines Frames in den Frequenzraum. In verschiedenen Bild- und Videokompressionsstandards wie bspw. JPEG, MPEG-1 und MPEG-2 wird eine zweidimensionale DCT der Größe 8×8 Pixel verwendet. Ein Block der Größe 8×8 Pixel wird in einen 8×8 DCT-Block transformiert. Dabei entsteht ein Block mit einem Gleichanteil (DC-Wert) sowie horizontalen und vertikalen Frequenzen (AC-Werte).

Da die DCT die Eigenschaft hat, dass sich ein Großteil der sichtbaren Informationen in wenigen niedrigen Frequenzen sammelt, schließen wir diese von den Veränderungen durch das Wasserzeichenverfahren aus. Dadurch erwarten wir eine signifikante Verbesserung der Transparenz. Dieser Vorgang kann auch mit einer Hochpaß-Filterung des Wasserzeichenmusters verglichen werden. Abbildung 5.1 zeigt einen 8×8 DCT-Block mit DCT-Koeffizienten, die im Zick-Zack-Modus angeordnet sind. Grau eingefärbt sind mögliche nicht verwendete Koeffizienten eines 8×8 DCT-Blocks. In diesem Fall schließen wir den DC-Wert (Index 0) und die 5 niedrigsten AC-Werte (Indizes 1-5) vom Markierungsprozess aus. In Abschnitt 5.1.5 werden wir zeigen, dass sich bei einer Nicht-Markierung von bis zu 6 DCT-Koeffizienten das Wasserzeichen immer noch als sehr robust erweist.



0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Abbildung 5.1: Nicht verwendete DCT-Koeffizienten

Abbildung 5.2 zeigt einen 8×8 Luminanzblock vor (links) und nach der Markierung (Mitte und rechts). Im mittleren Block wurden sämtliche Frequenzen modifiziert während im rechten Block der DC-Wert und die niedrigsten 5 AC-Werte vom Markierungsprozess ausgeschlossen wurden. Wie man aus den dargestellten Helligkeitswerten erkennen kann verursacht die Markierung aller Frequenzen ein stärkeres Rauschen, als bei Ausschluss der niedrigsten Frequenzen. Durchschnittlich wurden im mittleren Block die Helligkeitswerte um 6,44% verändert während im rechten Block die Werte durchschnittlich um 2,71% verändert wurden. Die Unterschiede der Helligkeitswerte sind in Abbildung 5.2 nicht auf den ersten Blick erkennbar. In aufeinanderfolgenden Videoframes können diese Unterschiede jedoch sichtbare Störungen hervorrufen.

5.1.2 Verbesserung der Robustheit

Bereits in [Fri97] stellt Fridrich fest, dass die Auslesequalität des Wasserzeichens im Frequenzraum besser als im Ortsraum ist. Um eine Verbesserung der Auslesequalität zu erreichen, verwenden wir statt der Korrelation im Ortsraum [Dit00] die normalisierte Kreuzkorrelation im Frequenzraum. Der Korrelationskoeffizient $C(B_j^*, P_j)$ eines Musters P_j mit einem potentiell markierten Block B_j^* wird wie folgt berechnet:

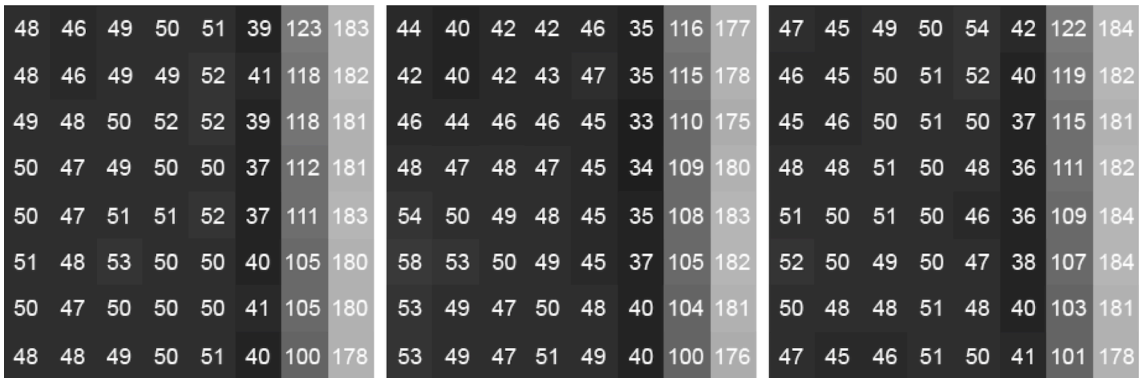


Abbildung 5.2: 8×8 Luminanzblock vor (links) und nach der Markierung (Mitte und rechts)

$$C(B_j^*, P_j) = \frac{\sum b_{ji}^* \cdot p_{ji}}{\sqrt{\sum {b_{ji}^*}^2} \cdot \sqrt{\sum p_{ji}^2}}. \tag{5.1}$$

In Gleichung (5.1) repräsentieren b_{ji}^* und p_{ji} jeweils die für die Markierung verwendeten DCT-Koeffizienten von B_j^* und P_j.

In Abschnitt 5.1.5 werden wir darüber hinaus untersuchen, ob durch die Vergrößerung des Wasserzeichenmusters eine weitere Verbesserung der Robustheit erreicht werden kann. Durch eine Vergrößerung der Wasserzeichenmuster erwarten wir, dass die Muster stärker mit dem zu untersuchenden markierten Inhalt korrelieren. Da das in [Dit00] vorgestellte visuelle Modell auf Luminanzblöcken der Größe 8×8 Pixel arbeitet verwenden wir Muster mit einer Seitenlänge, die durch 8 teilbar ist. Das Muster wird in 8×8 Pixel große Teilmuster unterteilt und diese auf den zu markierenden Inhalt aufgetragen. Im Ausleseprozess wird die durchschnittliche Korrelation aller Teilmuster mit ihren korrespondierenden Blöcken des Frames berechnet. Dabei wird wiederum das visuelle Modell angewandt und potentiell nicht markierte Blöcke von der Analyse ausgeschlossen.

5.1.3 Verbesserung der Kapazität

Zur Erhöhung der Kapazität wenden wir ein Verfahren an, das in [TSL09] vorgestellt wird. Das für Bilder entwickelte Wasserzeichen ist eine Kombination zweier Verfahren, die in [KR00] und [OP98] vorgestellt wurden. In diesem Bildwasserzeichenverfahren repräsentiert jedes Muster nicht nur ein Bit sondern eine binäre Sequenz der Länge N. Diese Sequenz ist Teil der Wasserzeichennachricht. Sollen durch das Muster 8 Bit der Wasserzeichennachricht repräsentiert werden, so benötigen wir 2^N also 256 verschiedene Muster. Im Ausleseprozess wird überprüft, mit welchem der 256 Muster der markierte Block am besten korreliert.

In Abschnitt 5.1.5 evaluieren wir diese Variation des Verfahrens hinsichtlich ihrer

Robustheit und Kapazität.

5.1.4 Verteilung der Wasserzeichennachricht

Eine Möglichkeit, eine Nachricht auf verschiedene Frames zu verteilen, stellen wir in [TSL09] vor. Der Aufbau der Nachricht wird in Gleichung (5.2) dargestellt. Sie wird dazu zunächst in Teilnachrichten gleicher Länge unterteilt (*Payload*). Jeder Teilnachricht wird eine eindeutige binäre Sequenz vorangestellt (*Sync*). Diese Sequenz ordnet der Teilnachricht ihre Position in der Gesamtnachricht zu. So besteht auch nach Verlust einzelner Frames die Möglichkeit die gesamte Nachricht wiederherzustellen. Die eindeutige binäre Sequenz kann eine feste oder variable Länge haben. Eine feste Länge hat den Vorteil, dass die Länge der Gesamtnachricht im Auslesesprozess als zusätzliche Information nicht bekannt sein muss, während eine variable Länge Vorteile in Bezug auf die Kapazität bringt. Zur Erhöhung der Auslesegenauigkeit kann jede Teilnachricht und/oder die Gesamtnachricht mit einer Fehlererkennung versehen werden. In der vorliegenden Implementierung verwenden wir den Cyclic Redundancy Check (*CRC*) für jede Teilnachricht.

$$Message = Sync \| Payload \| CRC \tag{5.2}$$

5.1.5 Evaluierung

Für die Evaluierung des Verfahrens verwenden wir ein MPEG-2 Testvideo des Instituts für Rundfunktechnik (IRT), welches uns im Rahnen des EU-Projekts „porTiVity“ zur Verfügung gestellt wurde. Es hat eine Framerate von 25 Frames pro Sekunde, eine Bitrate von 9.200 kBit/s und eine Auflösung von 720×576 Pixeln, die auf ein Seitenverhältnis von $16 : 9$ gespreizt werden. Es enthält Videosequenzen mit verschiedenen Charakteristiken:

- 1 - Menschenmenge: Viele Personen, kein Kameraschwenk, kein Schnitt, 252 Frames
- 2 - Eishockey: Kein Schnitt, viele Lichteffekte (Blitzlichter), viele Kameraschwenks, Mischung aus großer homogener Fläche (Eis) und texturierter Fläche (Publikum), 1802 Frames
- 3 - Jogger: Viele Schnitte, wenige Kameraschwenks, Mischung aus homogenen Flächen (Himmel) und stark texturierten Flächen (Boden), 1745 Frames
- 4 - Fußball: Wenige Schnitte, wenige Schwenks, große homogene Fläche (Rasen), 646 Frames
- 5 - Publikum: Kein Schnitt, langer Schwenk, stark texturierte Fläche, 680 Frames

- 6 - Musical: Wenige Schnitte, viele Lichteffekte, viele Personen, wenige Kameraschwenks, 1013 Frames
- 7 - Hubschrauberflug: Lange Kameraschwenks, viele Gebäude, wenige Schnitte, Mischung aus homogenen Flächen (Himmel) und stark texturierten Flächen (Gebäude), 734 Frames

Für die Evaluierung wurde das Video mit zwei Nachrichten mit einer Länge von 16 Bit markiert (Nachricht 1 = 1010101001010101, Nachricht 2 = 0101010110101010). Diese werden im folgenden Video 1 und Video 2 genannt. Somit sollte sichergestellt werden, dass an allen Positionen sowohl Nullen als auch Einsen korrekt eingebettet und wieder ausgelesen werden können. Pro Frame wurde eine Teilnachricht nach der in Abschnitt 5.1.4 beschriebenen Methode gebildet. Die Teilnachricht bestand aus der Synchronisierungssequenz (1 Bit), den Nachrichtenbits (8 Bit) und der Prüfsumme (4 Bit). Jede Teilnachricht wurde eine Sekunde lang wiederholt. Innerhalb von 4 Sekunden wurde also die komplette Nachricht einmal eingebettet. Anschließend wurde das markierte Video in MPEG-4/AVC konvertiert. Das komprimierte Video hat die gleiche Auflösung und Framerate wie das Originalvideo mit einer Bitrate von 1.500 kBit/s.

Im Ausleseprozess wurden Video 1 und Video 2 sekundenweise analysiert. Dazu wurden die gültigen Teilnachrichten betrachtet. Gültige Teilnachrichten sind solche, bei denen die Synchronisierungssequenz und die Nachrichtenbits durch die ausgelesene Prüfsumme verifiziert werden konnten. War die Mehrheit der gültigen Teilnachrichten in der Sekunde korrekt, so wurde diese Sekunde als korrekt anerkannt. Bei den folgenden Testergebnissen betrachten wir den Anteil der Sekunden, bei denen sowohl aus Video 1 als auch aus Video 2 die korrekte Teilnachricht ausgelesen wurde. Dieser Anteil wird im folgenden als Ausleserate bezeichnet.

Einfluss des Ausleseverfahrens auf die Robustheit

Im ersten Teil der Evaluierung untersuchen wir die in Abschnitt 5.1.2 aufgestellte These, dass durch das Auslesen des Wasserzeichens im Frequenzbereich die Ausleseergebnisse verbessert werden können. Abbildung 5.3 zeigt die Testergebnisse über das gesamte Video. Im linken Teil des Diagramms stellen wir die Ausleseverfahren im Frequenz- und Ortsbereich direkt nach der Markierung gegenüber. Im rechten Teil geschieht die Gegenüberstellung nach der Kompression. Es ist gut zu erkennen, dass die Ausleserate für das gesamte Video nach der Markierung identisch bleibt. Sie verbessert sich nach der Kompression minimal von 91,3% auf 92,4%.

In Abbildung 5.4 vergleichen wir die Ausleserate der beiden Ausleseverfahren in den einzelnen Szenen. Hier fällt auf, dass die Verbesserungen ausschließlich in der Szene „Jogger“ erzielt wurden. Sie verbessert sich nach der Kompression von 78,9% auf 83,1%. Bei allen anderen Szenen bleibt sie identisch.

Fazit: Durch die Umstellung des Ausleseverfahrens vom Orts- auf den Frequenzbe-

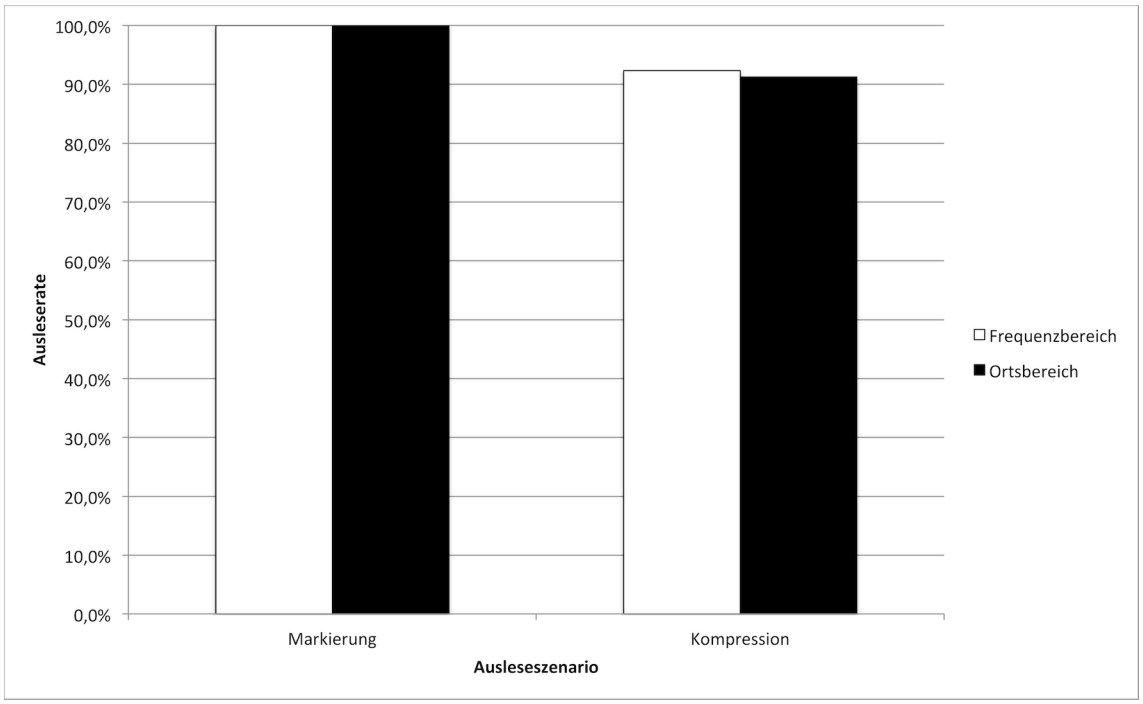


Abbildung 5.3: Analyse des Ausleseverfahrens (Gesamtvergleich)

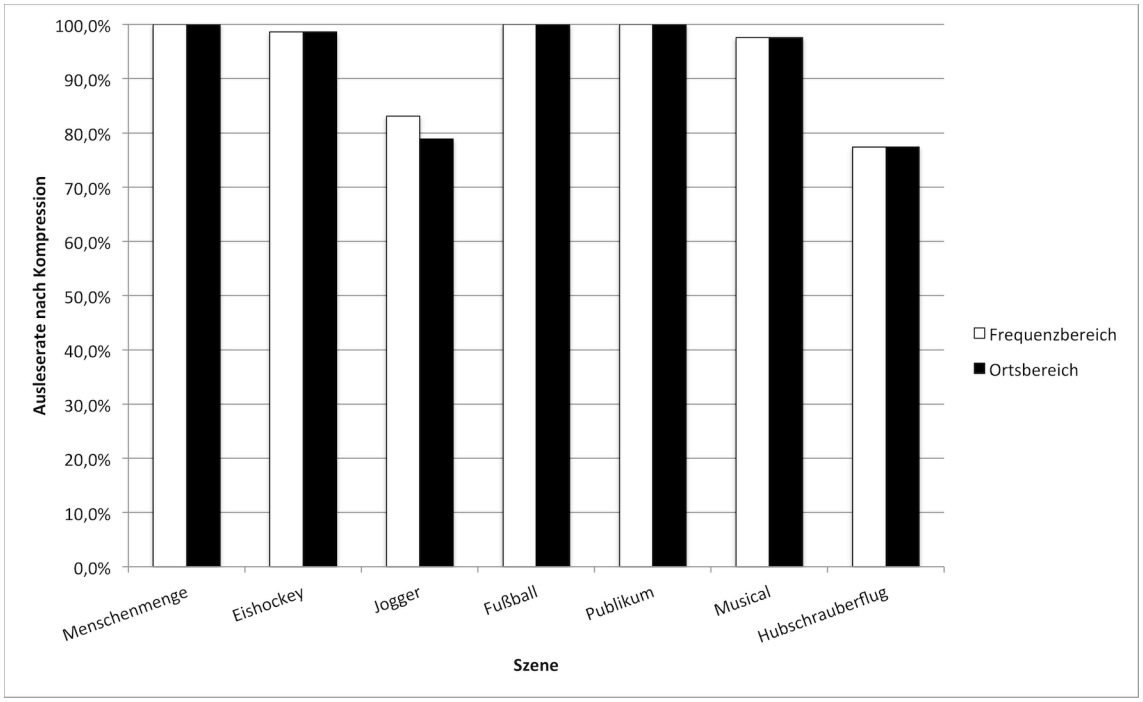


Abbildung 5.4: Analyse des Ausleseverfahrens nach Kompression (Szenenvergleich)

reich konnten minimale Verbesserungen erzielt werden, die sich auf nur eine Szene konzentrierten. Es kann sich also auch um ein statistisches Artefakt handeln. Die Robustheit des Verfahrens ist bei beiden Ausleseverfahren hoch.

Einfluss der ausgelassenen DCT-Koeffizienten auf die Robustheit

In Abschnitt 5.1.1 zeigten wir, dass durch das Auslassen von DCT-Koeffizienten bei der Markierung eines Videos die Transparenz verbessert werden kann. In diesem Abschnitt untersuchen wir die Auswirkung dieser Auslassung auf die Robustheit des Verfahrens. Bei den Tests markierten wir die ersten 3, 6, 10 und 15 DCT-Koeffizienten nicht. Damit wurde die gleiche Anzahl von horizontalen und vertikalen Frequenzen von der Markierung ausgeschlossen. In Abbildung 5.5 wird die Ausleserate über das gesamte Video nach der Markierung und nach der Kompression gegenübergestellt. Es fällt auf, dass die Markierung in allen Fällen erfolgreich verlief. Die Ausleserate beträgt hier 100%. Das Auslassen der DCT-Koeffizienten hat jedoch Auswirkungen auf die Ausleserate nach der verlustbehafteten Kompression. Ist sie bei 3 und 6 Koeffizienten mit 98,9% und 92,4% noch hoch, so sinkt sie bei 10 und 15 ausgelassenen Koeffizienten auf 77,5% und 47,3% ab. Mit weniger markierten Koeffizienten korrelieren demnach die Muster nicht mehr stark genug mit den markierten Blöcken, so dass eine gute Ausleserate nicht mehr gewährleistet ist.

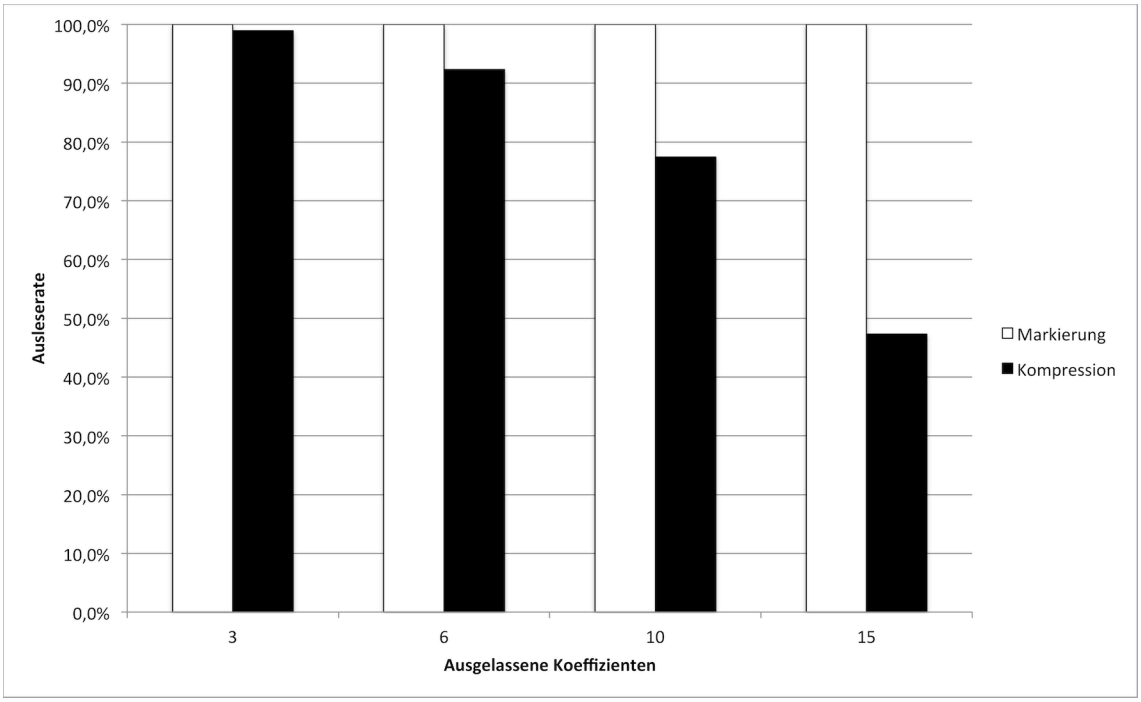


Abbildung 5.5: Analyse ausgelassener Koeffizienten (Gesamtvergleich)

In Abbildung 5.6 schlüsseln wir die Testergebnisse nach der Kompression nach Szenen auf. Hier ist zu erkennen, dass die Ausleserate insbesondere in den Szenen „Publikum“, „Musical“ und „Hubschrauberflug“ bei mehr ausgelassenen Koeffizienten stark absinkt. Erreichten wir bei 3 ausgelassenen Koeffizienten in der Szene „Hubschrauberflug“ noch eine Ausleserate von 96,8% so sinkt diese bei 15 ausgelassenen Koeffizienten auf bis zu 6,5%. Demgegenüber bleibt die Ausleserate in der Szene „Menschenmenge“ in allen Fällen konstant bei 100%. Die unterschiedlichen Ausleseraten können abhängig von der Charakteristik der jeweiligen Szenen sein. Es kann beispielsweise möglich sein, dass in einigen Szenen viele Blöcke eine zu starke Ten-

denz in Richtung einer Bitbelegung (bspw. 0) aufwiesen, so dass eine Markierung mit der anderen Bitbelegung (bspw. 1) nicht mehr robust möglich war. Hier müssten jedoch weitere Untersuchungen Aufschluss über die Ursache geben.

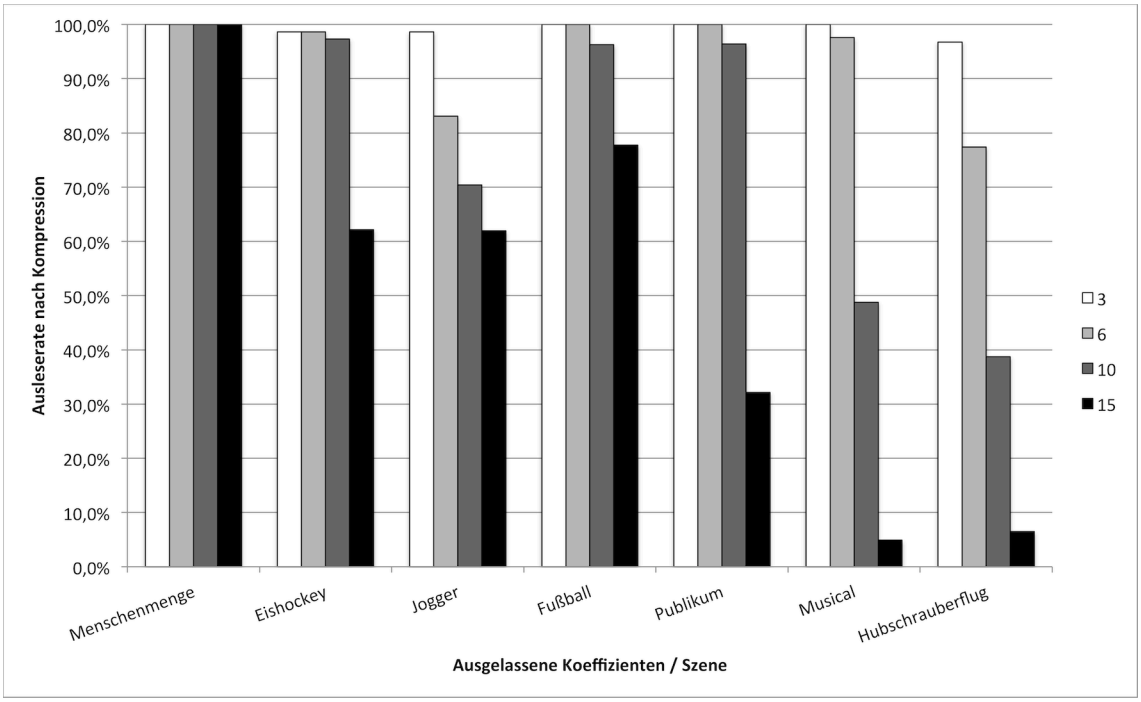


Abbildung 5.6: Analyse ausgelassener Koeffizienten nach Kompression (Szenenvergleich)

Fazit: Je weniger Koeffizienten markiert werden, desto größer sind die Auswirkungen auf die Robustheit des Verfahrens. Mit 3 bzw. 6 von der Markierung ausgelassenen Koeffizienten erreichen wir eine hohe Ausleserate von über 90%. In den folgenden Experimenten verwenden wir die letztere Einstellung, da mit 3 ausgelassenen Koeffizienten stark sichtbare Artefakte im Video feststellbar waren.

Einfluss der Blockgröße auf die Robustheit

In diesem Abschnitt untersuchen wir, wie sich die Vergrößerung der Muster und Blöcke auf die Robustheit des Verfahrens auswirken. Wir hatten die These aufgestellt, dass durch eine Vergrößerung der Blöcke eine höhere Korrelation zwischen den Wasserzeichenmustern und dem Inhalt und damit eine verbesserte Robustheit erreicht werden sollte. In Abbildung 5.7 stellen wir die Ausleseraten verschiedener Block- und Mustergrößen nach Markierung und verlustbehafteter Kompression gegenüber. Es wird deutlich, dass mit zunehmender Größe die Ausleserate abnimmt. Mit einer Blockgröße von 32×32 und 64×64 war eine korrekte Markierung des Videos nicht mehr möglich. Nach der Markierung erreichten wir hier eine Ausleserate von 98,2% und 94,2%. Dies setzte sich nach der verlustbehafteten Kompression fort. Hier erreichten wir Ausleseraten von 84,7% und 69,1%. Demgegenüber war die Ausleserate bei kleineren Blöcken und Mustern 96,4% und 92,4%.

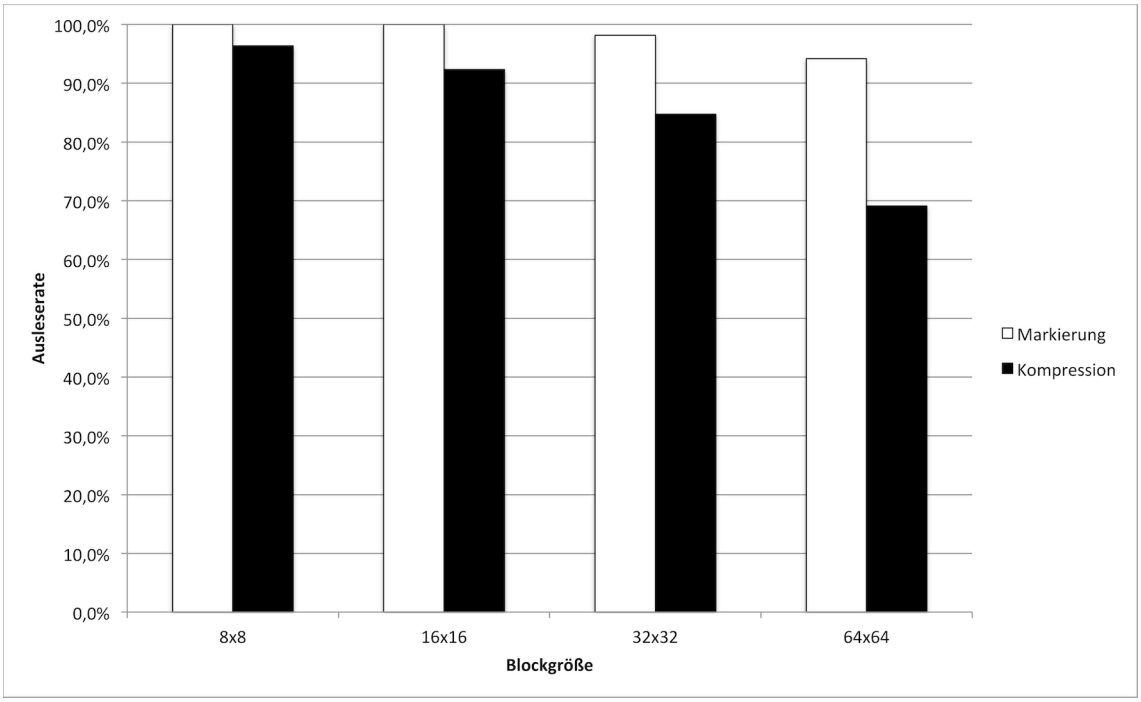


Abbildung 5.7: Analyse der Blockgröße (Gesamtvergleich)

Wiederum schlüsseln wir die Ausleseraten nach der Kompression in Abbildung 5.8 nach Szenen auf. Hier wird deutlich, dass bei der Szene „Hubschrauberflug“ die Ausleserate bei höherer Block- und Mustergröße am stärksten absinkt. Ähnliches ist bei den Szenen „Jogger“ und „Eishockey“ zu beobachten. Insbesondere ist bei letzterer ein deutliches Absinken der Ausleserate bei einer Block- und Mustergröße von 64×64 zu erkennen. Dies kann zum einen an der Charakteristik der markierten Szenen liegen, welche bereits im vorherigen Abschnitt als Grund für eine geringere Robustheit vermutet wurde. Zum anderen zieht eine größere Blockgröße eine geringere Redundanz nach sich. Die Nachricht kann also nicht mehr so oft im Frame eingebettet werden. Lehnt dann das visuelle Modell zusätzlich mehr Blöcke ab, so kann das Video nicht mehr korrekt markiert werden. Da die drei Szenen viele glatte Flächen enthalten kann die Kombination aus größeren Blöcken und der Ablehnung durch das visuelle Modell als Erklärung für eine geringere Robustheit in Frage kommen.

Fazit: Die Einführung größerer Block- und Mustergrößen wirkte sich nicht in erwünschter Weise auf die Robustheit des Verfahrens aus. Durch die Experimente wurde gezeigt, dass kleinere Block- und Mustergrößen besser zur Markierung geeignet sind.

Ermittlung der maximalen Kapazität

Im ersten Teil zur Ermittlung der maximalen Kapazität analysieren wir die in Abschnitt 5.1.3 vorgestellte Einbettungs- und Auslesestrategie. Wir untersuchen hier-

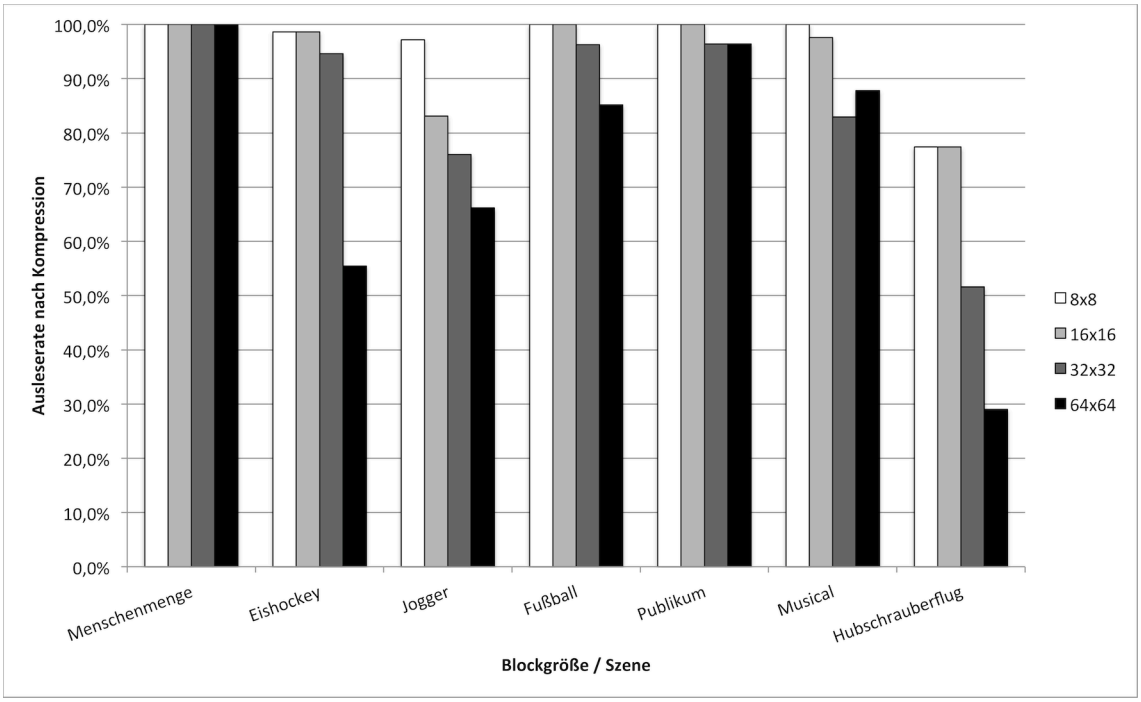


Abbildung 5.8: Analyse der Blockgröße nach Kompression (Szenenvergleich)

bei, wieviele Bits pro Block eingebettet werden können um gleichzeitig eine ausreichende Robustheit zu erzielen. In den Experimenten verwendeten wir eine Block- und Mustergröße von 16 × 16 Pixel, eine Prüfsumme der Länge 4 und eine Synchronisationssequenz der Länge 1. Die Länge der eingebetteten Teilnachricht war ein Vielfaches von 8.

Abbildung 5.9 zeigt die Testergebnisse für das gesamte Video. Wiederum wurden die Ausleseraten nach der Markierung und nach der Kompression ermittelt. Getestet wurden Einstellungen von 1 bis 7 Bits pro Block. Es ist erkennbar, dass die Ausleseraten nach der Markierung gut sind. Sie liegen im Bereich zwischen 99,3% und 100,0%. Das bedeutet, dass in fast allen Fällen das gesamte Video korrekt markiert werden konnte. Anders verhält es sich nach der verlustbehafteten Kompression. Mit zunehmender Kapazität nimmt die Ausleserate ab. Erreichten wir mit 1 Bit pro Block noch eine Ausleserate von 86,9%, so sank sie bei 4 Bits pro Block bereits auf 59,0%. Bei 7 Bits pro Block erreichten wir nur noch eine Ausleserate von 44,2% nach verlustbehafteter Kompression.

Die Aufschlüsselung nach Szenen in Abbildung 5.10 zeigt, in welchen Szenen die höchsten Verluste in der Ausleserate zu verzeichnen waren. Hier fällt auf, dass die Szenen „Publikum“, „Musical“ und „Hubschrauberflug“ hauptverantwortlich für die niedrigen Ausleseraten sind. Akzeptable Ausleseraten konnten nur für 1 (zwischen 64,5% und 92,9%) und 2 Bits pro Block (zwischen 41,9% und 75,0%) erreicht werden. Auch hier gibt es möglicherweise wieder einen Trend zu einem der Wasserzeichenmuster, der durch den Einbettungsprozess nicht mehr korrigiert werden konnte. Es bleibt hier eine Untersuchung offen, ob durch die Verwendung anderer Schlüssel das Wasserzeichen eine höhere Robustheit ausweist. Ist dies der Fall so würde die These

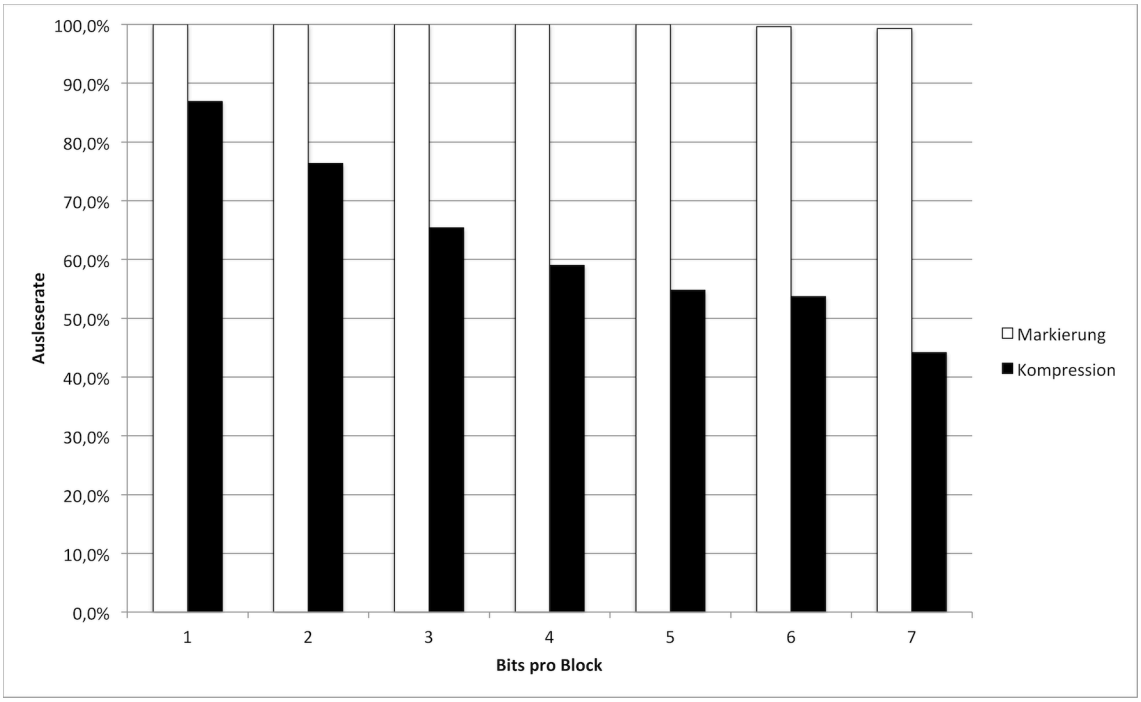


Abbildung 5.9: Analyse der Einbettung mehrerer Bits pro Block (Gesamtvergleich)

vom Trend in Richung eines Wasserzeichenmusters bestätigt werden.

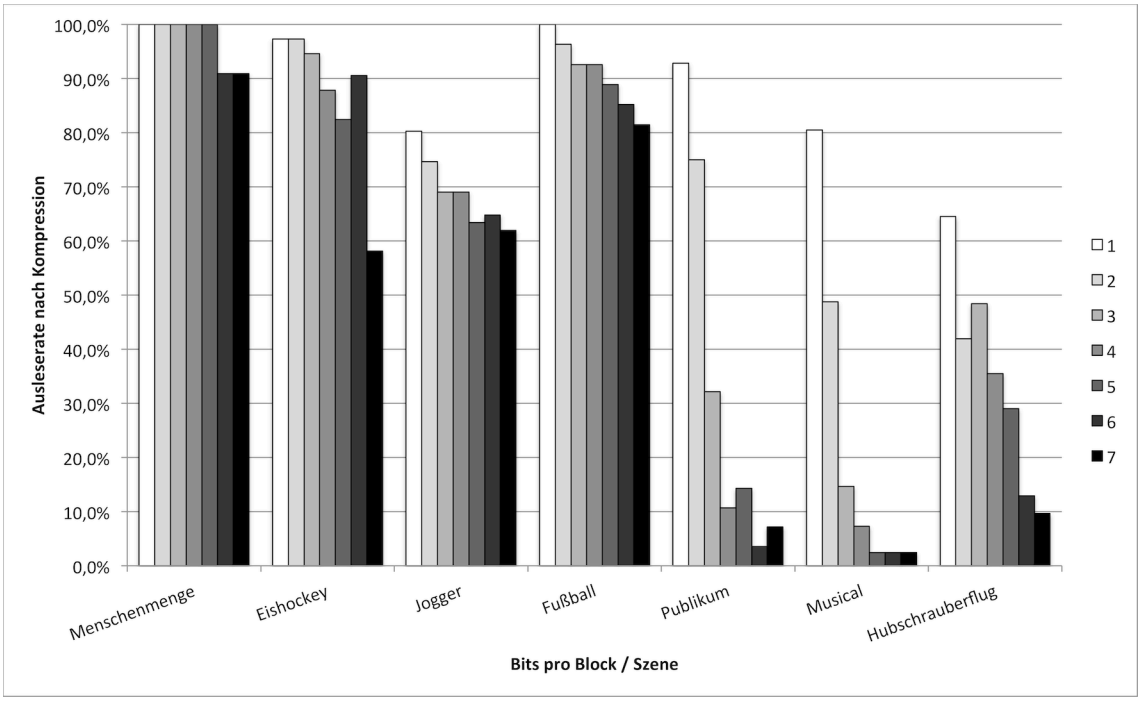


Abbildung 5.10: Analyse der Einbettung mehrerer Bits pro Block nach Kompression (Szenenvergleich)

Fazit: Durch die Änderung der Strategie, mehrere Bits pro Block einzubetten, konnte die maximale Kapazität pro Sekunde von 8 auf 16 Bit erhöht werden. Die Auslese-

raten schwanken in Abhängigkeit der Szene zwischen 41,9% und 100,0%.

5.2 DCT-Koeffizienten-Relationsverfahren

In [TVDS04] stellen wir ein Verfahren vor, das zum einen an das Verfahren von Langelaar et al. [LLB99] und zum anderen an das Verfahren von Zhao und Koch [ZK95] angelehnt ist. Da es ursprünglich für MPEG kodierte Videos entwickelt wurde, arbeitet es auf quantisierten DCT-Koeffizienten in DCT-Blöcken der Größe 8×8. Es kann jedoch auch auf nicht-quantisierte DCT-Koeffizienten übertragen werden.

5.2.1 Methode

Das Verfahren arbeitet in zwei Schritten. Im ersten Schritt wird ein Bit m_i einer Wasserzeichennachricht $M = \{m_0, \ldots, m_{n_m-1}\}$ in eine Blockgruppe eingebracht. Im zweiten Schritt wird dann m_i zusätzlich in jeden Block der Gruppe eingebracht.

Sei I ein Bild oder Videoframe und n_b die Anzahl der zur Markierung verwendeten DCT-Blöcke. In der Vorbereitungsphase werden die verwendeten DCT-Blöcke pseudo-zufällig gemischt. Der Mischvorgang wird mit einem geheimen Schlüssel K kontrolliert. Anschließend bilden wir Blockgruppen $G_i = \{b_{i,0}, \ldots, b_{i,n_G-1}\}$, wobei jede Gruppe G_i insgesamt n_G DCT-Blöcke enthält. Der Durchschnitt $A_{b_{i,j}}$ eines Blockes $b_{i,j}$ aus einer Gruppe G_i wird wie folgt berechnet:

$$A_{b_{i,j}} = \frac{1}{64} \sum_{c=0}^{63} \left| \theta_{b_{i,j},c} \right| \tag{5.3}$$

Analog zu [LLB99] bezeichnet $\theta_{b_{i,j},c}$ den DCT-Koeffizienten c (in Zick-Zack-Anordnung) des Blockes $b_{i,j}$. Der Durchschnitt A_{G_i} einer Gruppe G_i wird wie folgt berechnet:

$$A_{G_i} = \frac{1}{n_G} \sum_{j=0}^{n_G-1} A_{b_{i,j}} \tag{5.4}$$

Wir bilden auf Basis der Gleichungen (5.3) und (5.4) zwei Mengen:

$$\begin{aligned} S_> &= \{b_{i,j} | A_{b_{i,j}} > A_{G_i}\} \\ S_< &= \{b_{i,j} | A_{b_{i,j}} < A_{G_i}\} \end{aligned} \tag{5.5}$$

Soll Gruppe G_i eine 1 repräsentieren, so muss $S_>$ nach der Markierung mehr Elemente enthalten als $S_<$. Gilt $m_i = 0$ so ist diese Mengenrelation umgekehrt. Gilt $m_i = 1$ und ist die Bedingung noch nicht erfüllt, so wird der Block $b_{i,j}$ in $S_<$ mit dem höchsten Durchschnitt $A_{b_{i,j}}$ so lange modifiziert, bis $A_{b_{i,j}} > A_{G_i}$ gilt. Dazu werden mittels K ausgewählte mittlere Frequenzen von $b_{i,j}$ erhöht. Gleichzeitig wird der Durchschnitt eines anderen Blockes in $S_<$ erniedrigt, so dass A_{G_i} konstant bleibt. Für den Fall $m_i = 0$ wird die Prozedur in umgekehrter Reihenfolge durchgeführt. Der Block $b_{i,j}$ aus $S_>$ wird so lange modifiziert, bis $A_{b_{i,j}} < A_{G_i}$ gilt. Um A_{G_i} konstant zu halten wird der Durchschnitt eines weiteren Blockes aus $S_>$ in gleichem Umfang erhöht.

Um eine höhere Redundanz der Nachricht pro Blockgruppe zu erreichen, wird im zweiten Schritt in jeden Block $b_{i,j}$ zusätzlich ein Bitmuster eingebracht, das den Wert von m_i repräsentiert. Dazu werden pseudo-zufällig vier Koeffizienten des Blockes $b_{i,j}$ derart modifiziert, dass sie mit der folgenden Gleichungsfolge einen dreistelligen Referenzvektor $V = \{v_0\, v_1\, v_2\}$ bilden:

$$\begin{aligned} v_0 &= |c_0 - c_1| \ mod\ 2 \\ v_1 &= |c_0 - c_2| \ mod\ 2 \\ v_2 &= |c_0 - c_3| \ mod\ 2 \end{aligned} \tag{5.6}$$

Es gilt:

$$V = \begin{cases} 101 & falls \quad m_i = 1 \\ 011 & falls \quad m_i = 0 \end{cases} \tag{5.7}$$

Abbildung 5.11 zeigt ein Beispiel für das Einbringen eines Bitmusters. Der Referenzvektor, der eingebracht werden soll, ist $V = 101$. Im unmarkierten Block würden die Koeffizienten nach Gleichung (5.6) den Vektor $V = 010$ repräsentieren. Um nur minimale Veränderungen vorzunehmen, wird im vorliegenden Fall c_0 angepasst.

Es gilt zu beachten, dass das Einbringen des Wasserzeichens in die vier Koeffizienten nicht den Blockdurchschnitt $A_{b_{i,j}}$ verändern darf. Wird also ein Koeffizient erhöht, so muss gleichzeitig ein anderer, nicht beteiligter Koeffizient verringert werden.

Im Ausleseprozess werden zunächst wieder die Blockgruppen gebildet. Anschließend wird für jede Blockgruppe G_i die zugehörigen Blockmengen $S_>$ und $S_<$ gebildet. Enthält $S_>$ mehr Elemente als $S_<$, so wird aus der Blockgruppe G_i eine 1 ausgelesen. Enthält $S_<$ mehr Elemente als $S_>$, so wurde potentiell eine 0 eingebettet. Darüber hinaus wird in jedem Block $b_{i,j}$ der Vektor V nach Gleichung (5.6) bestimmt. Die Ergebnisse des Gruppenverhältnisses und der Blockverhältnisse werden gewichtet, um die Auslesequalität der Nachricht zu bestimmen.

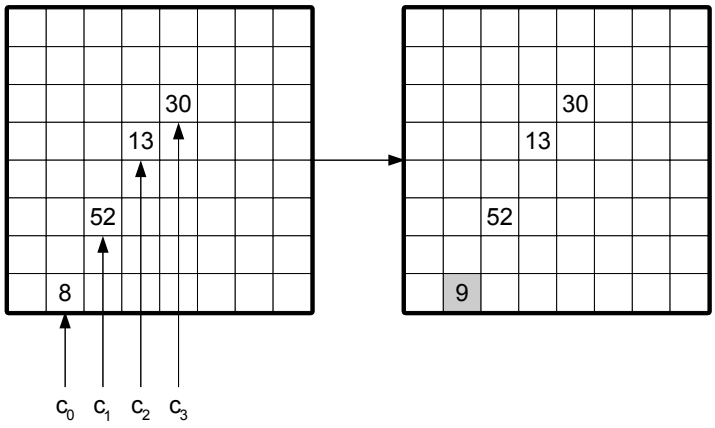


Abbildung 5.11: Beispiel für ein Blockmuster

5.2.2 Evaluierung

Zur Evaluierung der Robustheit wurden zehn MPEG-1 Videos unterschiedlicher Länge und mit unterschiedlichen Charakteristiken markiert und anschließend manipuliert. Die Videos hatten eine Länge zwischen 5 und 120 Sekunden. Die Videoauflösung betrug bei allen Videos 352 × 288 Pixel bei einer Bitrate von 575 kBit/s, 1.150 kBit/s bzw. 2.300 kBit/s und einer Framerate von 25 Frames pro Sekunde. Es wurden nur die I-Frames markiert. Die Kapazität war dabei 60 Bit/Frame. Auf den Videos wurden folgende Manipulationen ausgeführt:

- Fester Wert addiert: Dieser Angriff addiert einen festen Wert auf die quantisierten Luminanz-DCT-Koeffizienten. Folgende Werte wurden verwendet: -5, -3, +3, +5. Bei einer Veränderung von 5 wurden bereits signifkante Störungen registriert.

- Weißes Rauschen addiert: Ähnlich zum vorherigen Angriff werden Werte auf die quantisierten Luminanz-DCT-Koeffizienten addiert. Dabei werden jedoch keine festen sonder pseudozufällig ausgewählte Werte verwendet. Der Maximalwert des Rauschens betrug 3 bzw. 5.

- Reenkodierung: Die Videos wurden mit Bitraten von 575 kBit/s, 1.150 kBit/s bzw. 2.300 kBit/s reenkodiert. Es kam also teilweise zu einer verlustbehafteten Kompression.

- Skalierung: Die Videos wurden auf die neue Auflösung 480×352 Pixel skaliert. Es ist zu beachten, dass die Videos für den Ausleseprozess nicht auf ihre Originalauflösung zurückgeführt wurden, so dass es zu einer Desynchronisierung des Wasserzeichens kam.

Die Ergebnisse sind in Abbildung 5.12 dargestellt. Alle Videos konnten korrekt markiert werden. Das Verfahren zeigt eine gute Robustheit gegenüber einer erneuten

Enkodierung mit MPEG-1 (durchschnittlich 96,81%), dem Hinzufügen fester Werte (durchschnittlich 99,54%) und dem Hinzufügen von Weißem Rauschen (durchschnittlich 100%). Obwohl es mit der Skalierung auf eine höhere Auflösung zu einer Desynchronisierung des block-basierten Wasserzeichens kam, betrug die Ausleserate immer noch durchschnittlich 65,87%. Das Verfahren weist eine hohe Transparenz bei einer gleichzeitig hohen Kapazität auf. Aufgrund dieser Eigenschaften würde das Verfahren sehr gut geeignet für den Einsatz in inhaltsfragilen Videowasserzeichen sein. Allerdings wurde seine Entwicklung nicht fortgeführt, da man sich auf die Weiterentwicklung des Verfahrens von Fridrich und Dittmann konzentriert hat.

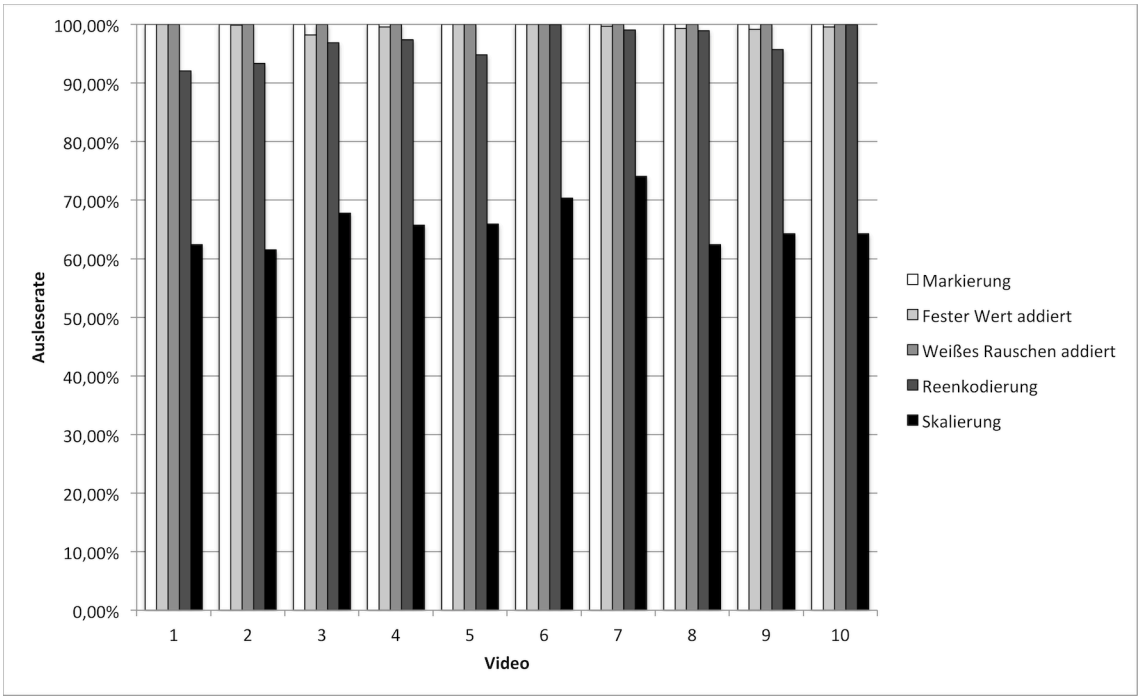


Abbildung 5.12: Evaluierung der Robustheit des DCT-Koeffizienten-Relationsverfahrens

5.3 Zusammenfassung

Mit den in diesem Kapitel vorgestellten Verfahren und Ergebnissen wurden die Grundlagen für die inhalts-fragilen Wasserzeichen gelegt. Die Verfahren zeigen eine gute Robustheit gegenüber verlustbehafter Kompression und Formatumwandlung bei gleichzeitig guter Transparenz und Kapazität auf. In Kapitel 7.2 werden wir darüber hinaus zeigen, dass Pixelfehler bzw. Objektmanipulationen keine Auswirkung auf die Auslesequalität unseres verbesserten Fridrich-Verfahrens haben.

Manipulationen an der Zeitachse (Entfernen von Frames und Szenen) haben aufgrund des in Abschnitt 5.1.4 eingeführten speziellen Aufbaus der Wasserzeichenteilnachrichten keinen Einfluss auf die Auslesequalität des Verfahrens von Fridrich.

Beide Verfahren sind nicht robust gegen Drehung, Skalierung und Cropping, wenn

keine Resynchronisierung des Materials erfolgt. Hier besteht Forschungsbedarf für die Zukunft um Verfahren zu entwickeln, die diese Eigenschaften bei gleichzeitig guter Transparenz und hoher Kapazität aufweisen. Weiterer Forschungsbedarf besteht in der Untersuchung der Robustheit gegenüber dem Einsatz von Weichzeichenfiltern und Helligkeits- bzw. Kontrastanpassungen.

Im folgenden verwenden wir unser verbessertes Fridrich-Verfahren für die Markierung mit inhaltsbeschreibenden Merkmalen.

Kapitel 6

Inhaltsfragile Merkmale

Neben der Auswahl des zugrundeliegenden robusten Wasserzeichens ist die Auswahl des inhaltsbeschreibenden Merkmals von entscheidender Bedeutung für ein inhaltsfragiles Wasserzeichen. Die Herausforderung bei der Entwicklung eines solchen Merkmals besteht in der möglichst kompakten Beschreibung des Bild-/Videoframeinhalts. Dabei muss beim Schutz digitaler Videos mit Wasserzeichen darauf geachtet werden, dass sowohl einzelne Videoframes als auch die Zeitachse mit Hilfe des Merkmals geschützt werden. Im Rahmen dieser Arbeit wurden vier Verfahren entwickelt, die Merkmale aus Videos extrahieren und diese auf binäre Vektoren fester Länge abbilden. Dabei wird zwischen zwei Arten von Merkmalen unterschieden:

- Merkmale mit nicht-inhaltlichen Abhängigkeiten: Diese Merkmale verwenden Eigenschaften des Videos, die nicht direkt vom Inhalt abgeleitet werden, d.h. sie beziehen sich auf Eigenschaften des spezifischen Videoformats bzw. auf Eigenschaften, die keinen direkten Rückschluss auf den sichtbaren Inhalt zulassen. Die Vertreter dieser Kategorie werden in den Abschnitten 6.1 und 6.2 vorgestellt.

- Merkmale mit inhaltlichen Abhängigkeiten: Diese Merkmale verwenden sichtbare Eigenschaften des Videos. Diese sichtbaren Eigenschaften können Kanten (siehe [DSS99]), markante Punkte oder Objektbeschreibungen sein. In den Abschnitten 6.3 und 6.4 stellen wir zwei Verfahren vor, die auf Basis von markanten Punkten arbeiten. Sie ziehen eine deutlich komplexere Berechnung nach sich als Merkmale mit nicht-inhaltlichen Abhängigkeiten. Wir erhoffen uns jedoch dadurch eine Verbesserung der Robustheit und Sensitivität.

6.1 Verwendung der Energiedifferenz

Das von uns in [DTS04] vorgestellte Merkmal beschreibt nicht-inhaltliche Abhängigkeiten zwischen Koeffizienten von DCT-Blöcken. Es basiert auf der folgenden

Annahme von Lin et al. [LC00] über das Verhältnis von DCT-Koeffizienten in JPEG-Bildern:

Seien $F_p(v)$ und $F_q(v)$ Koeffizienten zweier DCT-Blöcke p und q an der Positon v in Zick-Zack-Anordnung mit $v \in \{1, \ldots, 64\}$. Sei ferner $\Delta F_{p,q}(v) = F_p(v) - F_q(v)$. Das Verhältnis zwischen $F_p(v)$ und $F_q(v)$ wird durch die Funktion $\Phi_p(v)$ wie folgt definiert:

$$\Phi_p(v) = \begin{cases} 1, & \Delta F_{p,q}(v) \geq 0 \\ 0, & \Delta F_{p,q}(v) < 0 \end{cases} \tag{6.1}$$

Nach Lin et al. bleibt das durch $\Phi_p(v)$ beschriebene Verhältnis nach einer verlustbehafteten Kompression gleich. Der Beweis für diese These wird von den Autoren in [LC01] vorgestellt. Da eine Abbildung des Verhältnisses einzelner DCT-Koeffizienten auf einen Merkmalsvektor die Kapazität eines robusten Wasserzeichenverfahrens überschreiten würde, verwenden wir als Grundlage für die Generierung des Merkmalsvektors die in Kapitel 3.1.2 beschriebene Energiedifferenz von Langelaar et al. [LLB99]

6.1.1 Generierung des Merkmalsvektors und Einbettung

Sei $F = \{f_1, \ldots, f_N\}$ die Menge aller Videoframes in einem Video. Sei weiterhin $B_n = \{b_{n,1}, \ldots, b_{n,M}\}$ die Menge aller 8×8 DCT-Blöcke in einem Frame f_n. Für jeden Block $b_{n,m}$ sei ferner $C_{n,m} = \{c_{n,m,1}, \ldots, c_{n,m,64}\}$ die Menge seiner DCT-Koeffizienten in Zick-Zack-Anordnung.

Im ersten Schritt unterteilen wir $C_{n,m}$ in zwei Teilmengen:

- $C_{n,m,Merkmal} = \{c_{n,m,1}, \ldots, c_{n,m,S}\}$ sei die Menge der Koeffizienten, die zur Merkmalsberechnung verwendet werden
- $C_{n,m,Wasserzeichen} = \{c_{n,m,S+1}, \ldots, c_{n,m,64}\}$ sei die Menge der Koeffizienten, die zur Einbettung des Wasserzeichens verwendet werden

Im zweiten Schritt werden aus benachbarten Blöcken P gleich große Blockgruppen $G_{n,p}$ gebildet. Jede Blockgruppe enthält dabei M/P Blöcke.

Im dritten Schritt berechnen wir die partielle Energie $E_{Teil}(G_{n,p})$ jeder Blockgruppe $G_{n,p}$. Dazu wird Gleichung (3.5) wie folgt modifiziert:

$$E_{Teil}(G_{n,p}) = \sum_{b_{n,m}} \sum_{c_{n,m,s}} c_{n,m,s}^2 \tag{6.2}$$

mit $b_{n,m} \in G_{n,p}$ und $c_{n,m,s} \in C_{n,m,Merkmal}$.

Im vierten Schritt werden mittels eines geheimen Schlüssels K Blockgruppenpaare zwischen Frame f_n und seinem vorhergehenden Frame f_{n-1} gebildet. Ein Vektorbit $v_{n,o}$ des Merkmalsvektors V_n wird aus einem Blockgruppenpaar (G_{n-1,o_1}, G_{n,o_2}) wie folgt gebildet:

$$v_{n,o} = \begin{cases} 1, & E_{Teil}(G_{n-1,o_1}) - E_{Teil}(G_{n,o_2}) \geq 0 \\ 0, & E_{Teil}(G_{n-1,o_1}) - E_{Teil}(G_{n,o_2}) < 0 \end{cases} \tag{6.3}$$

Der Merkmalsvektor V_n des Frames f_n ist definiert als die Konkatenation aller Vektorbits $v_{n,o}$.

Die Einbettung des Merkmalsvektors V_n in das Frame f_{n+1} erfolgt analog zu dem in Kapitel 3.1.2 vorgestellten Einbettungsverfahren von Langelaar et al. Dabei entspricht V_n der eingebetteten Nachricht M. Es muss gelten: $c_{min} \geq S$. Der Schwellwert c_{min} aus Gleichung 3.8, der bestimmt, bis zu welchem Koeffizienten maximal die AC-Koeffizienten gelöscht werden dürfen, muss also mindestens so groß sein wie S, der die Menge der Koeffizienten bestimmt, die zur Merkmalsberechnung verwendet werden. Das Merkmal darf also durch den Einbettungsprozess nicht gestört werden. Die Einbettung in das nachfolgende Frame ist notwendig, da inhalts-verändernde Maßnahmen gleichzeitig das Wasserzeichen beeinträchtigen könnten. Damit wäre eine Lokalisierbarkeit manipulierter Stellen nicht mehr möglich. Zusätzlich wird ein eindeutiger, kontinuierlich steigender Frameindex dem Merkmalsvektor angehängt, um Manipulationen an der Zeitachse zu erkennen. Manipulationen an der Zeitachse sind das Einfügen und Entfernen von Frames sowie die Änderung ihrer Abspielreihenfolge.

Abbildung 6.1 stellt das Verfahren zur Generierung des Merkmalsvektors noch einmal schematisch dar. Für die Frames f_{n-1} und f_n bilden wir Gruppen und berechnen die partiellen Energien. Mit dem Schlüssel K werden Gruppenpaare gebildet und die Verhältnisse der Gruppenpaare auf den Merkmalsvektor abgebildet. Der Merkmalsvektor V_n wird anschließend in das Frame f_{n+1} eingebettet.

In Abbildung 6.2 stellen wir ein Frame seinen partiellen Energien gegenüber. Wir berechneten dabei die Energie vom ersten bis einschließlich zum fünften DCT-Koeffizienten. Zur Verbesserung der Sichtbarkeit wurden die Energiewerte gespreizt. Ein dunklerer Block hat also relativ eine geringere Energie. Wie erkennbar ist konzentrieren sich die hohen Energien auf den linken, oberen Bereich des Bildes. Der untere, rechte Bereich ist stärker texturiert. Diese Texturen befinden sich jedoch in den mittleren und hohen Frequenzen, so dass die Energie bei den ersten fünf DCT-Koeffizienten in Relation zum Gesamtbild niedriger ist.

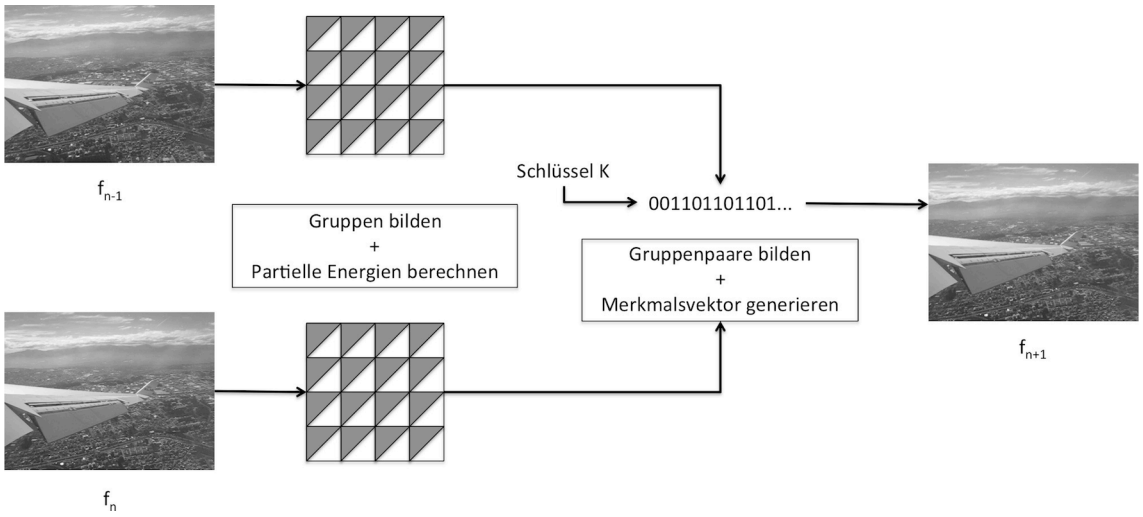


Abbildung 6.1: Merkmalsgenerierung für Energiedifferenzverfahren

Abbildung 6.2: Gegenüberstellung eines Frames und seiner zugehörigen Energiewerte

6.1.2 Auslesen des Merkmalsvektors und Verifikation

Im Verifikationsprozess wird zunächst der mittels eines Wasserzeichenverfahrens eingebettete Vektor V_n aus Frame f_{n+1} ausgelesen. Dies geschieht analog zu dem in Kapitel 3.1.2 beschriebenen Ausleseprozess von Langelaar et al. Aus Frame f_{n-1} und Frame f_n berechnen wir den Merkmalsvektor $\tilde{V_n}$ analog zum Einbettungsprozess. $\tilde{V_n}$ wird nach Gleichung (6.3) gebildet.

Zur Lokalisierung von möglichen Manipulationen werden V_n und $\tilde{V_n}$ verglichen. Unterscheiden sich $v_{n,o} \in V_n$ und $\tilde{v}_{n,o} \in \tilde{V_n}$, so können drei Fälle aufgetreten sein:

1. In Frame f_{n-1} wurde die Gruppe G_{n,o_2} manipuliert: Hier sollten die Fehler in f_{n-1} gehäuft auftreten. Da gleichzeitig der in f_{n-1} eingebettete Vektor V_{n-1} durch die Manipulation Fehler aufweisen könnte, würde somit auch f_{n-2} als fehlerhaft angezeigt werden. In f_{n-2} sollten jedoch die Fehler durch die Permutation der Markierungspositionen gestreut und nicht gehäuft auftreten.

2. In Frame f_n wurde die Gruppe G_{n-1,o_1} manipuliert: Hier sollten die Fehler in f_n gehäuft auftreten. Da gleichzeitig der in f_n eingebettete Vektor V_n durch die Manipulation Fehler aufweisen könnte, würde somit auch f_{n-1} als fehlerhaft angezeigt werden. In f_{n-1} sollten jedoch die Fehler durch die Permutation der Markierungspositionen gestreut und nicht gehäuft auftreten.

3. Sowohl Frame f_{n-1} als auch Frame f_n wurden manipuliert: In diesem Fall sollten sowohl gestreute als auch gehäufte Fehler in f_{n-1} und gehäufte Fehler in f_n auftreten. Dies leitet sich aus den beiden vorher beschriebenen Fällen ab. Zusätzlich erwarten wir gestreute Fehler in f_{n-2} aufgrund der Beschädigung des Vektors V_{n-1}, der in f_{n-1} eingebettet wurde.

6.1.3 Analyse des Merkmals

Die Analyse des Merkmals wurde im Hinblick auf verschiedene Aspekte durchgeführt. Wir unterscheiden in unserer Analyse zwischen inhalts-erhaltenden und inhalts-verändernden Maßnahmen. Für unsere Tests verwendeten wir das in Kapitel 5.1.5 vorgestellte Testvideo. Die untersuchten inhalts-erhaltenden Maßnahmen umfassten:

- Formatumwandlung (MPEG-2 nach MPEG-4 AVC) und anschließende verlustbehaftete Kompression mit zwei verschiedenen Bitraten (4.600 kBit/s und 1.500 kBit/s)

- Formatumwandlung (MPEG-2 nach MPEG-4 AVC) und Skalierung auf drei verschiedene Auflösungen (90%, 75% und 50% der Original-Auflösung)

Die untersuchten inhalts-verändernden Maßnahmen umfassten:

- Austauschen zweier Blöcke: Die Veränderungen wurden paarweise oben-links und unten-rechts bzw. oben-rechts und unten-links durchgeführt. Die Größe der veränderten Blöcke betrug 5% bzw. 10% der Gesamtfläche. Insgesamt werden also 10% bzw. 20% verändert.

- Entfernen eines Blockes: Die Farbwerte der einzelnen Pixel des betreffenden Blockes wurden durch den Farbmittelwert des Blockes ersetzt. Die Veränderungen wurden in allen vier Ecken durchgeführt. Die Größe der veränderten Blöcke betrug 5% bzw. 10% der Gesamtfläche.

- Einfügen eines Blockes: Der betreffende Block wurde durch einen weißen Block mit schwarzer Schrift ersetzt. Die Veränderungen wurden in allen vier Ecken durchgeführt. Die Größe der veränderten Blöcke betrug 5% bzw. 10% der Gesamtfläche.

Ausgewählte Beispiele dieser Maßnahmen sind in Abbildung 6.3 dargestellt. Im oberen linken Bild wurden der obere rechte und der untere linke Block ausgetauscht. Im oberen rechten Bild wurde der untere rechte Block durch seinen Farbmittelwert ersetzt. Im unteren Bild wurde der obere linke Block durch weißen Block mit schwarzer Schrift ersetzt um das Einfügen eines Blockes zu simulieren.

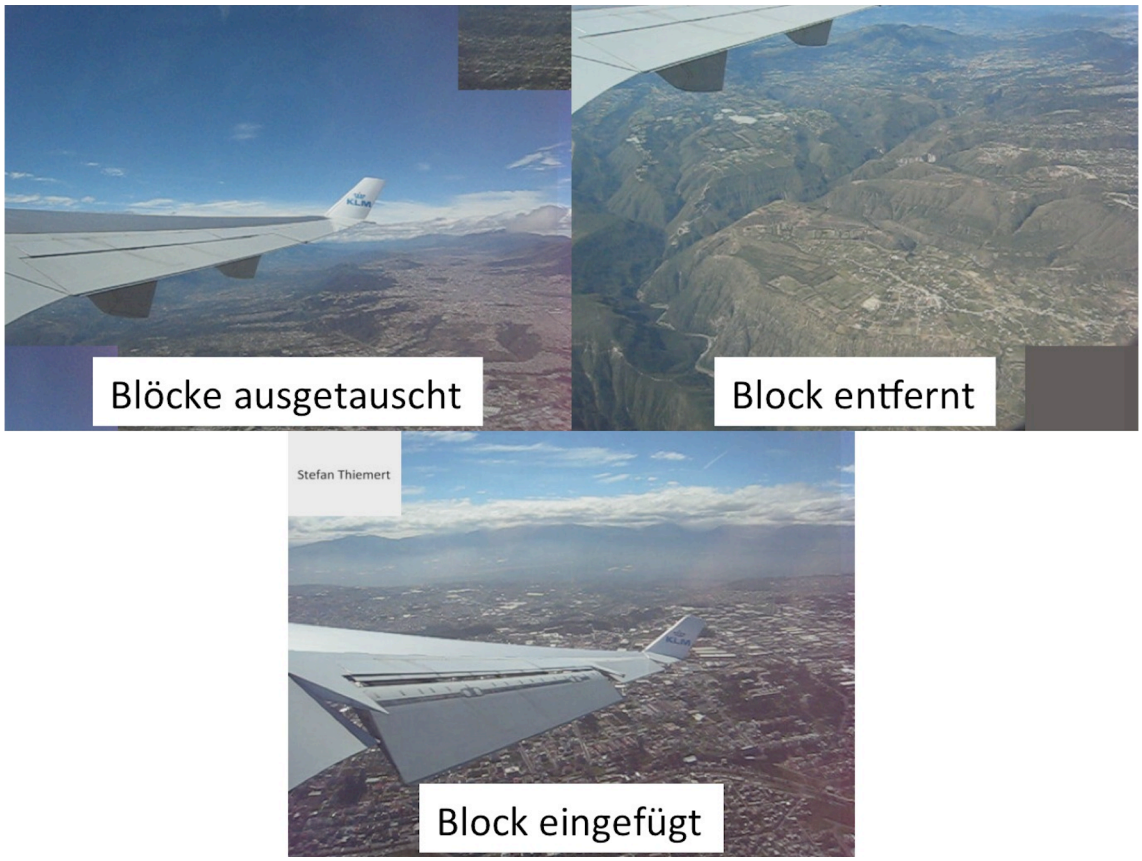


Abbildung 6.3: Beispiele für inhalts-verändernde Maßnahmen

Zur Generierung des Merkmals haben wir unterschiedliche Größen der Blockgruppen untersucht. Drei Auflösungen wurden getestet:

Auflösung in Blöcken	Auflösung in Pixeln	Vektorlänge
9x7	72x56	100 Bit
6x5	48x40	210 Bit
5x3	40x24	432 Bit

Tabelle 6.1: Getestete Auflösungen des Energiedifferenz-Merkmals

Zusätzlich untersuchen wir die Einführung eines Schwellwertes T, der die Gleichung (6.3) wie folgt modifiziert:

$$v_{n,o} = \begin{cases} 1, & E_{Teil}(G_{n-1,o_1}) - E_{Teil}(G_{n,o_2}) \geq T \\ 0, & E_{Teil}(G_{n-1,o_1}) - E_{Teil}(G_{n,o_2}) < -T \end{cases} \tag{6.4}$$

T kann entweder ein fester oder ein dynamischer Schwellwert sein. In unseren Untersuchungen verwendeten wir einen dynamischen Schwellwert, der wie folgt definiert wurde:

- Ist $E_{Teil}(G_{n-1,o_1})$ die höhere von beiden Energien, so wird der Schwellwert T über den prozentualen Anteil von $E_{Teil}(G_{n-1,o_1})$ definiert.
- Ist $E_{Teil}(G_{n,o_2})$ die höhere von beiden Energien, so wird der Schwellwert T über den prozentualen Anteil von $E_{Teil}(G_{n,o_2})$ definiert.

T ist also der prozentuale Anteil der höheren Energie.

Für die folgenden Testergebnisse wurde die Anzahl der Koeffizienten, die zur Berechnung des Merkmals verwendet wurden, variiert. Für den Parameter S aus Abschnitt 6.1.1 verwendeten wir die Werte 2, 5, 9, 14 und 20. Der Grund liegt darin, dass die gleiche Anzahl von horizontalen und vertikalen Frequenzen einbezogen werden sollte. Die folgenden Testergebnisse repräsentieren den jeweils besten Parametersatz für die jeweilige Auflösung. Der beste Parametersatz zeichnet sich durch das beste Verhältnis zwischen Robustheit und Sensitivität aus. Der beste Parametersatz ist also nicht unbedingt derjenige, der die beste Sensitivität aufweist, weil dies mit einem Rückgang der Robustheit einhergehen könnte. Ebenso könnte ein Parametersatz mit einer hervorragenden Robustheit wiederum eine mangelnde Sensitivität nach sich ziehen. Die Parametersätze aus Tabelle 6.2 wurden für die verschiedenen Vektorlängen verwendet.

Im Diagramm in Abbildung 6.4 wird die Total Rejection Rate (TRR) dargestellt. Sie repräsentiert den Anteil aller als verändert erkannter Blöcke in Relation zur Gesamtanzahl aller Blöcke. Es wird ersichtlich, dass mit kürzerer Länge des Vektors auch die TRR sinkt. Je kleiner also die Blockgruppen sind, desto fehleranfälliger wird das Merkmal, die TRR steigt an. Sehr gut zu erkennen ist, dass in allen drei Auflösungen das Merkmal bei allen fünf inhalts-erhaltenden Maßnahmen eine Fehlerrate von unter einem Prozent aufweist.

In den Abbildungen 6.5 und 6.6 stellen wir die Correct Rejection Rate (CRR) und die False Rejection Rate (FRR) dar. Die CRR repräsentiert den Anteil aller Frames, die als verändert erkannt wurden. Ein verändertes Frame gilt dann als erkannt, wenn mindestens einer der veränderten Blöcke vom Merkmal erkannt wurde. Demgegenüber repräsentiert die FRR den Anteil der Blöcke, die unkorrekt als verändert erkannt wurden, in Relation zur Gesamtanzahl aller Blöcke. Die Ergebnisse wurden, wie bereits in Abbildung 6.4, hinsichtlich der Länge des Merkmalsvektors gruppiert. Drei Erkenntnisse können wir aus dem Diagramm ableiten:

1. Je größer die Veränderung war, desto besser wurde die Veränderung erkannt. Wurden Blöcke mit einer Fläche von 5% der Gesamtfläche verändert, so wurden durchschnittlich 30,26% aller Manipulationen korrekt erkannt (210 Bit). Wurden dagegen Blöcke mit einer Fläche von 10% der Gesamtfläche verändert, so stieg die durchschnittliche Erkennungsrate auf 61,62% (ebenfalls 210 Bit).

2. Veränderungen an Blockpaaren (Austausch von Blöcken) wurden besser erkannt als an Einzelblöcken. Betrug die Länge des Merkmalsvektors 210 Bit so wurden durchschnittlich 81,02% aller veränderten Frames, bei denen Blockpaare verändert wurden, richtig erkannt. Im Fall von manipulierten Einzelblöcken betrug die Erkennungsrate durchschnittlich 40,92% (bei eingefügten Blöcken) bzw. 15,88% (bei entfernten Blöcken).

3. Der Merkmalsvektor mit einer Länge von 210 Bit erzielte die besten Ergebnisse. Durchschnittlich betrug hier die Erkennungsrate 45,94%, wobei sie zwischen 6,66% (Blöcke von 5% Größe entfernen) und 93,61% (Blöcke von 10% Größe austauschen) schwankt. Gleichzeitig ist die durchschnittliche FRR bei den inhalts-verändernden Maßnahmen mit 3,37% am niedrigsten im Vergleich zu Merkmalsvektoren mit einer Länge von 432 Bit und 100 Bit.

In den Abbildungen 6.7, 6.8 und 6.9 zeigen wir die Auswirkungen des Schwellwerts T auf Merkmalsvektoren der Länge 210 Bit. Dafür wurden die Parametersätze aus Tabelle 6.3 verwendet. Wiederum sind zwei Erkenntnisse aus den Ergebnissen abzuleiten:

1. Die Einführung von T hatte keine signifikanten Auswirkungen auf die Total Rejection Rate bei den inhalts-erhaltenden Maßnahmen. Aus Abbildung 6.7 wird ersichtlich, dass bei der stärksten Veränderung, der Kompression auf 1.500 kBit/s, die TRR bei allen untersuchten Schwellwerten immer noch bei unter einem Prozent liegt.

2. Mit steigendem Schwellwert T steigt die Correct Rejection Rate bei den inhalts-verändernden Maßnahmen. Aus Abbildung 6.8 wird ersichtlich, dass bei einem Schwellwert von 0% die durchschnittliche CRR bei 38,88% liegt. Demgegenüber erhöht sich bei einem Schwellwert von 12% die durchschnittliche Erkennungsrate auf 45,94%.

Fazit: Die Einführung von T hat die Ergebnisse noch einmal deutlich verbessert.

Vektorlänge	Schwellwert S	Schwellwert T
100 Bit	5	12%
210 Bit	5	12%
432 Bit	2	12%

Tabelle 6.2: Parametersätze für die verschiedenen Vektorlängen

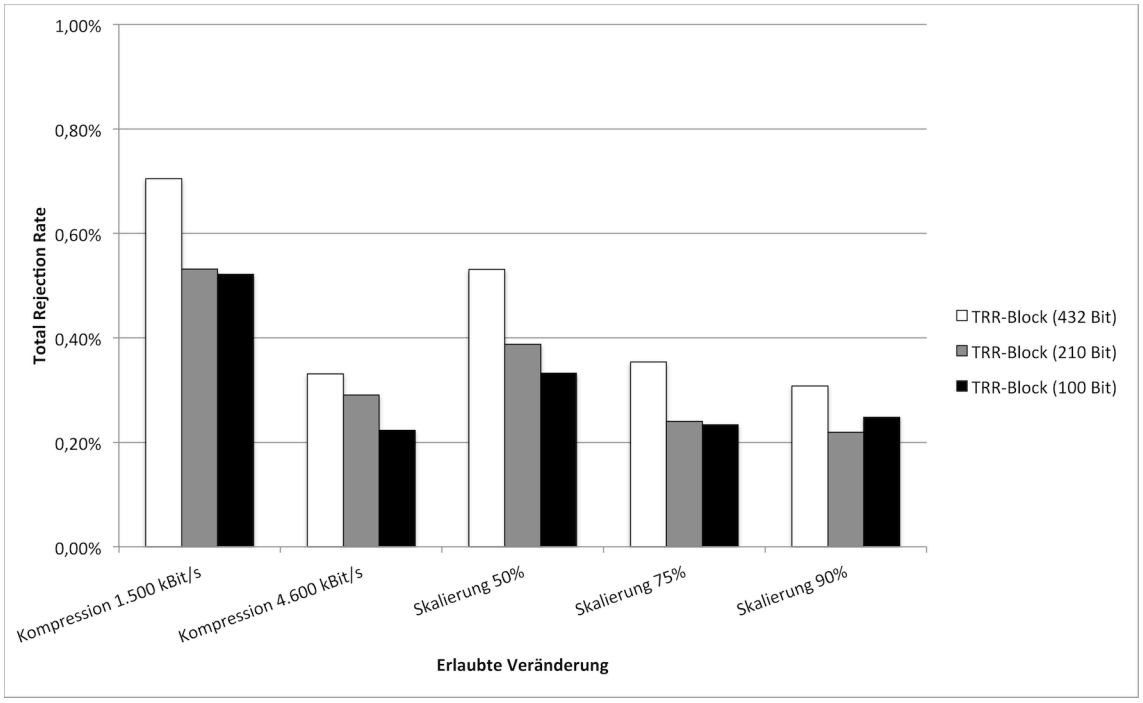


Abbildung 6.4: TRR des Energiedifferenzverfahrens gruppiert nach Länge des Merkmalsvektors

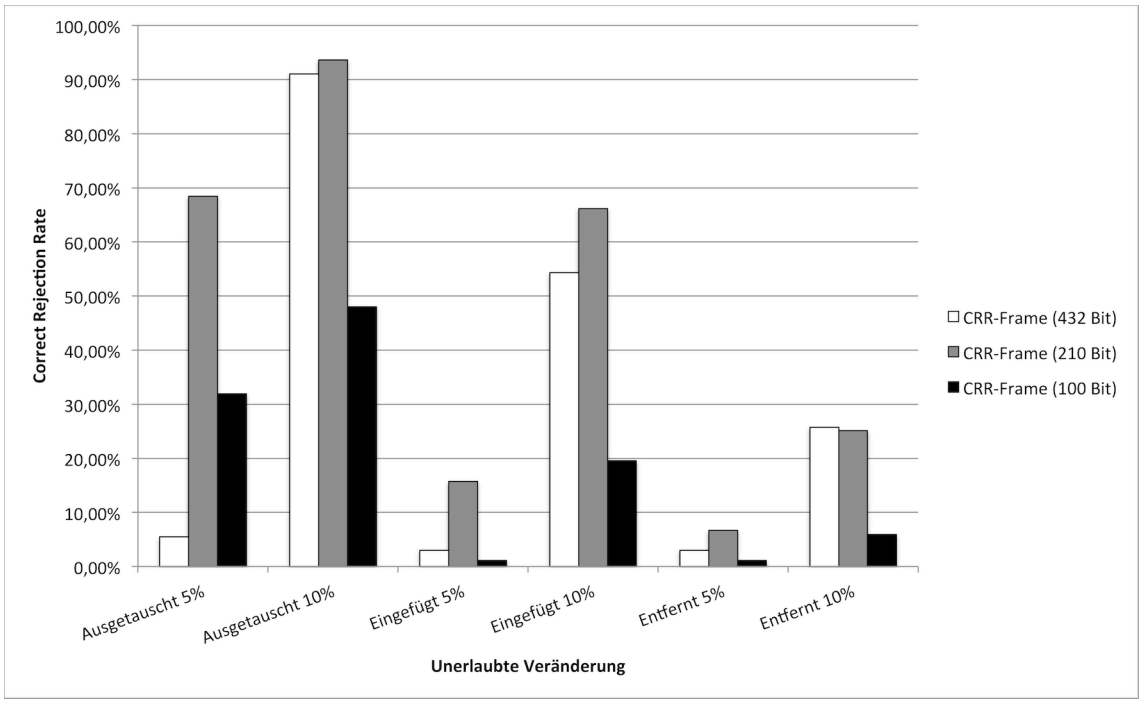


Abbildung 6.5: CRR des Energiedifferenzverfahrens gruppiert nach Länge des Merkmalsvektors

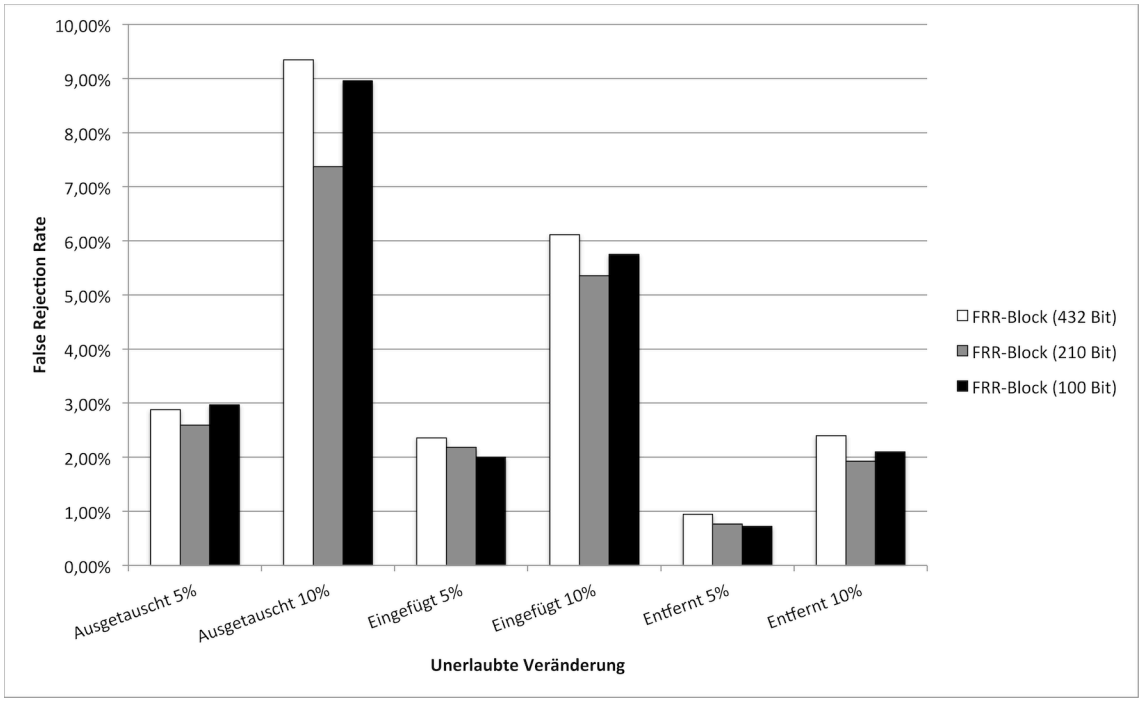


Abbildung 6.6: FRR des Energiedifferenzverfahrens gruppiert nach Länge des Merkmalsvektors

Schwellwert T	Schwellwert S
0%	9
6%	2
12%	5

Tabelle 6.3: Parametersätze für die verschiedenen Werte des Schwellwertes T

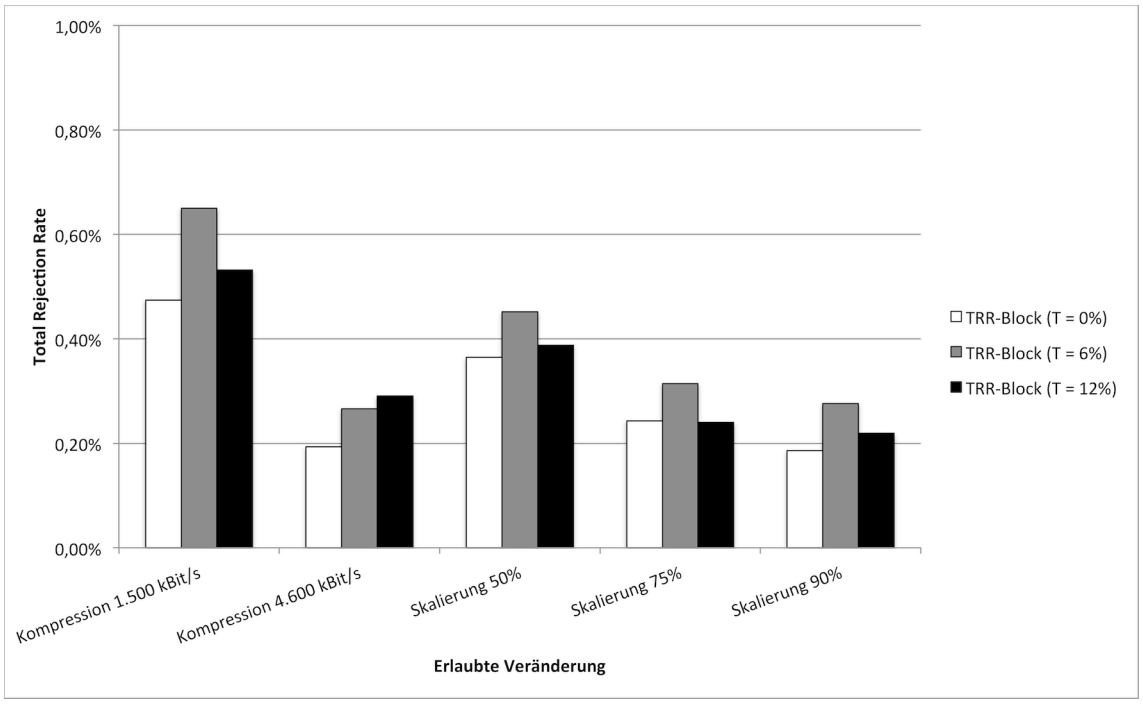


Abbildung 6.7: TRR des Energiedifferenzverfahrens gruppiert nach Schwellwert T

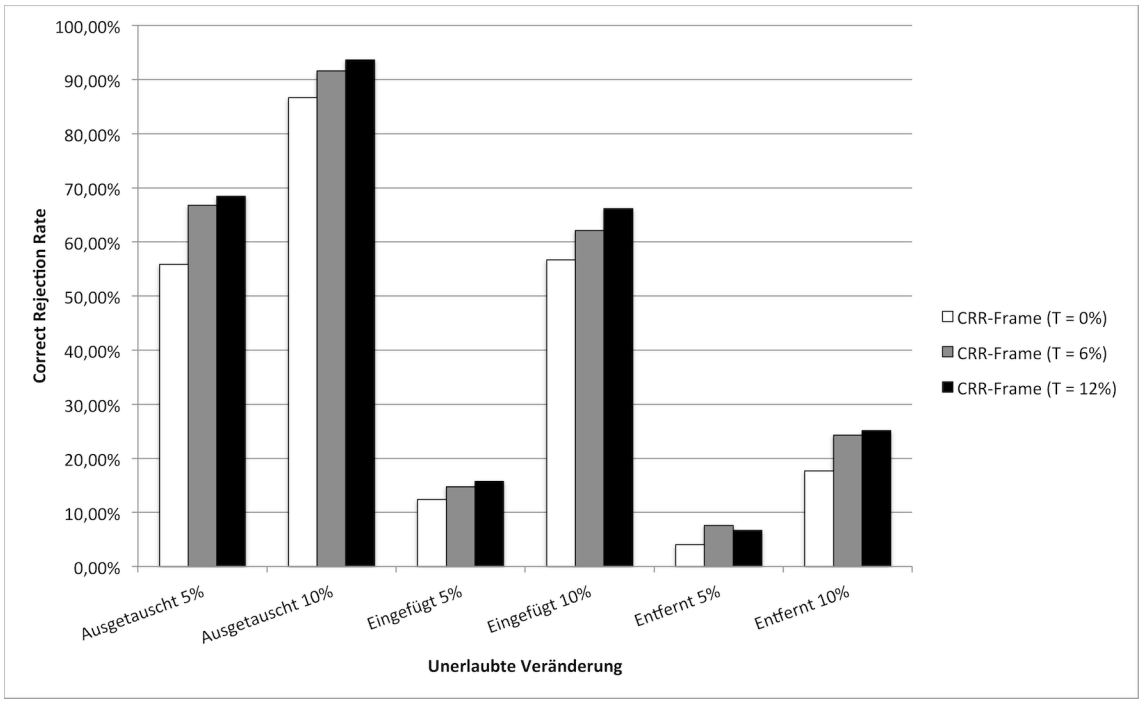


Abbildung 6.8: CRR des Energiedifferenzverfahrens gruppiert nach Schwellwert T

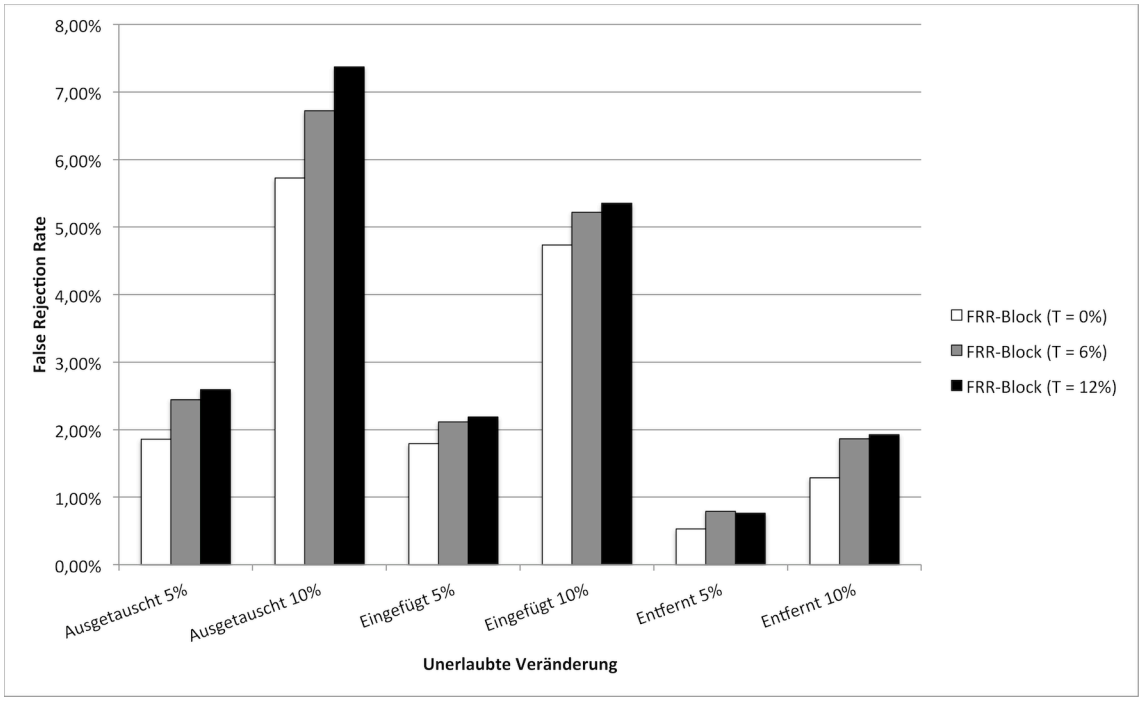


Abbildung 6.9: FRR des Energiedifferenzverfahrens gruppiert nach Schwellwert T

6.2 Verwendung der Grauwert-Entropie

Ein weiteres Merkmal, das nicht-inhaltliche Abhängigkeiten betrachtet und für den Einsatz in Integritätswasserzeichen geeignet ist, ist die Entropie der Grauwerte und wurde von uns in [TSS06] vorgestellt. Die Entropie wurde erstmals von C. E. Shannon in [Sha48] vorgestellt und bezeichnet in der Informationstheorie die Informationsdichte eines Zeichensystems.

Sei $Z = \{z_1, \ldots, z_U\}$ ein Alphabet mit U Elementen und $P = \{p_1, \ldots, p_U\}$ die Menge der zugehörigen Wahrscheinlichkeiten mit $\sum_{p_u \in P} = 1$. Dann ist die Entropie $H(P)$ definiert als:

$$H(P) = -\sum_{u=1}^{N} p_u \cdot \log_2 p_u \tag{6.5}$$

Aus Sicht der Informationstheorie bedeutet dies, dass durchschnittlich $H(P)$ Bits benötigt werden, um ein Element aus Z zu kodieren [CT91]. Für Kompressionsalgorithmen repräsentiert die Entropie also die durchschnittlich erreichbare Kompressionsrate.

In unserem Fall ist $Z = \{0, 1, \ldots, 255\}$ die Menge der Grauwerte mit $U = 256$ Elementen. Sei $B_n = \{b_{n,1}, \ldots, b_{n,M}\}$ die Menge aller Blöcke in einem Frame $f_n \in F$, wobei $F = \{f_1, \ldots, f_N\}$ die Menge aller Videoframes in einem Video ist. Sei weiterhin P die Wahrscheinlichkeitsverteilung aller Grauwerte in Block $b_{n,m}$ mit $P = \{p_{n,m,0}, \ldots, p_{n,m,255}\}$ und $\sum_{j=0}^{255} p_{n,m,j}$. Dann ist $H(b_{n,m})$ die Entropie der Grauwerte in Block $b_{n,m}$. Dazu wird Gleichung (6.5) folgendermaßen modifiziert:

$$H(b_{n,m}) = -\sum_{j=0}^{255} p_{n,m,j} \cdot \log_2 p_{n,m,j} \tag{6.6}$$

6.2.1 Generierung des Merkmalsvektors und Einbettung

Um den Merkmalsvektor $V_n = v_{n,1} \| \ldots \| v_{n,M}$ eines Frames f_n zu generieren wird zunächst das Frame auf eine feste Größe skaliert. Damit wird eine feste Länge des Merkmalsvektors erreicht. In der Praxis verwenden wir eine Frameauflösung von 128×128 Pixeln. Danach wird jeder Block $b_{n,m}$ durch die folgenden Schritte vorverarbeitet:

1. Anwendung eines Tief-Pass-Filters auf den Block. Damit werden die mittleren und hohen Frequenzen von der Merkmalsgenerierung ausgeschlossen. Diese Frequenzen werden im späteren Einbettungsprozess verwendet.

2. Anwendung eines Weichzeichen-Filters, um das gefilterte Bild zu glätten.

3. Quantisierung der Grauwerte auf QF Werte, um die Anzahl der Grauwerte zu reduzieren und damit eine höhere Robustheit zu erreichen.

4. Berechnung der Entropie $H(b_{n,m})$ nach Gleichung (6.6).

In einem weiteren Schritt werden die Blöcke in Gruppen $G_n = \{G_{n,1}, \ldots, G_{n,O}\}$ unterteilt. Ein Vektorbit $v_{n,m} \in V_n$ wird durch eine der folgenden Bedingungen gebildet:

- Ist $H(b_{n,m})$ innerhalb seiner Gruppe $G_{n,o}$ ein Maximum, dann wird sein zugehöriges Vektorbit $v_{n,m}$ auf 1 gesetzt.
- Ist $H(b_{n,m})$ innerhalb seiner Gruppe $G_{n,o}$ größer als die Hälfte aller $H(b_{n,m})$ in $G_{n,o}$, dann wird sein zugehöriges Vektorbit $v_{n,m}$ auf 1 gesetzt.
- Erfüllt $H(b_{n,m})$ keine der vorherigen Bedingungen, so wird sein zugehöriges Vektorbit $v_{n,m}$ auf 0 gesetzt.

Für die Einbettung des Merkmalsvektors verwenden wir die mittleren und hohen Frequenzen. Dazu eignen sich alle in den Kapiteln 3.1.2 und 5 vorgestellten Verfahren. Die Einstellungen müssen so vorgenommen werden, dass die niedrigen Frequenzen, die zur Merkmalsgenerierung verwendet wurden, nicht durch den Einbettungsprozess verändert werden. Es ist wieder darauf zu achten, dass der Merkmalsvektor V_n in das benachbarte Frame f_{n+1} eingebettet wird, um die Lokalisierbarkeit nach einer Veränderung des Inhalts gewährleisten zu können. Zusätzlich wird ein eindeutiger, kontinuierlich steigender Frameindex dem Merkmalsvektor angehängt, um Manipulationen an der Zeitachse zu erkennen.

Abbildung 6.10 stellt das Verfahren zur Generierung des Merkmalsvektors noch einmal schematisch dar. Das Frame f_n wird vorverarbeitet und in Gruppen unterteilt, deren jeweilige Entropie berechnet wird. Aus den Gruppen-Entropien wird der Merkmalsvektor generiert und unter der Kontrolle eines geheimen Schlüssels K in das Frame f_{n+1} eingebettet.

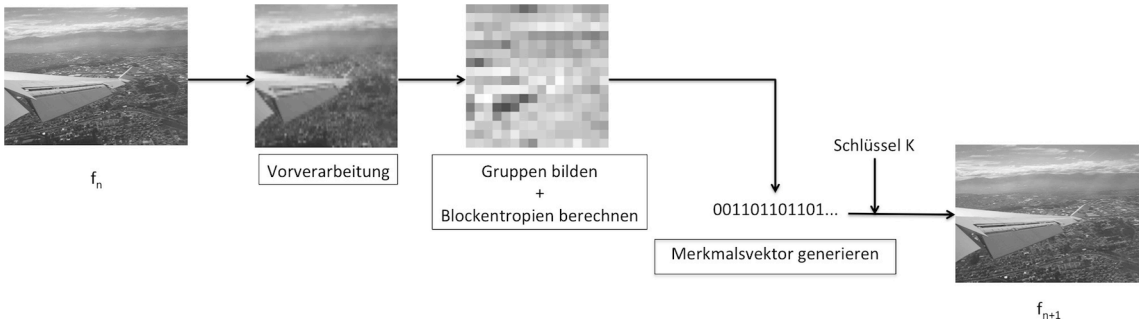


Abbildung 6.10: Merkmalsgenerierung für Entropieverfahren

Abbildung 6.11 stellt ein Frame beispielhaft seinen Block-Interestwerten gegenüber. Die Block-Interestwerte wurden zur besseren Sichtbarkeit gespreizt. Ein dunklerer Block verfügt also über eine geringere Entropie. Es ist zu erkennen, dass im Bereich der Tragflächen und des Himmels die Entropie deutlich geringer ist als beispielsweise im unteren rechten Bereich (erkennbar an den dunkleren Blöcken). Eine stärkere Texturierung hat also eine höhere Entropie zur Folge. Im Gegenzug hat eine geringere Texturierung folglich auch eine niedrigere Entropie zur Folge.

Abbildung 6.11: Gegenüberstellung eines Frames und seiner Entropie-Interestwerte

6.2.2 Auslesen des Merkmalsvektors und Verifikation

Im Ausleseprozess wird der Merkmalsvektor V_n aus dem Frame f_{n+1} ausgelesen. Darüber hinaus berechnen wir den aktuellen und möglicherweise modifizierten Merkmalsvektor $\tilde{V}_n$ aus Frame f_n. Stimmen die Bits von V_n und $\tilde{V}_n$ nicht überein, so ist von einer Manipulation des Videos auszugehen. Die aufgetretenen Manipulationen können lokalisiert und angezeigt werden. Wir unterscheiden drei Fälle von Manipulationen:

1. Wurde Frame f_n manipuliert, so sollten Fehler in f_n gehäuft auftreten, während Fehler in f_{n-1} gestreut auftreten sollten. Dieser Fall kann dann auftreten, wenn durch die Manipulationen der Merkmalsvektor V_{n-1}, der in Frame f_n eingebettet wurde, beschädigt wurde.

2. Wurde Frame f_{n+1} manipuliert, so sollten Fehler in f_n gestreut auftreten, wenn der Merkmalsvektor in V_n in f_{n+1} beschädigt wurde. Gleichzeitig sollten in Frame f_{n+1} Fehler gehäuft auftreten.

3. Wurden mehrere Frames manipuliert, so sollten in verschiedenen Frames die Fehler sowohl gehäuft als auch gestreut auftreten. Im Frame vor Beginn der

Manipulation sollten durch die Authentifizierung durch das beschädigte nachfolgende Frame die Fehler gestreut auftreten. Demgegenüber sollten im letzten Frame der Manipulation die Fehler nur gehäuft auftreten, da der authentifizierende Merkmalsvektor aus dem nachfolgenden Frame nicht beschädigt sein sollte.

In [TS10] stellen wir eine Möglichkeit vor um inhalts-erhaltende Maßnahmen von inhalts-verändernden Maßnahmen zu unterscheiden. Dazu nutzen wir die Zeitachse eines Videos aus. Wir vertreten die Annahme, dass inhalts-erhaltende Maßnahmen, wie verlustbehaftete Kompression oder Kontrastanpassung, zufällige Störungen im Merkmal hervorrufen. Um die Aussage des sichtbaren Inhaltes zu verändern müssen mehrere Frames hintereinander über einen längeren Zeitraum manipuliert werden. Um die Unterscheidung zu ermöglichen führen wir einen zeitlichen Filter ein. Tritt eine Störung im Merkmal mehrfach an der gleichen Position auf, so ist das ein Indiz für eine inhalts-verändernde Maßnahme. Abbildung 6.12 zeigt auf der linken Seite detektierte Veränderungen am Merkmal nach verlustbehafteter Kompression, also einer inhalts-erhaltenden Maßnahme. Oben abgebildet ist das Ergebnis vor und unten nach der Anwendung eines zeitlichen Filters. Wie gut zu erkennen ist, tritt die Störung zufällig auf und wird durch den Filter nahezu eliminiert. Anders verhält es sich mit einer inhalts-verändernden Maßnahme, die auf der rechten Seite abgebildet ist. Sie bleibt zu großen Teilen auch nach der Anwendung des zeitlichen Filters erhalten.

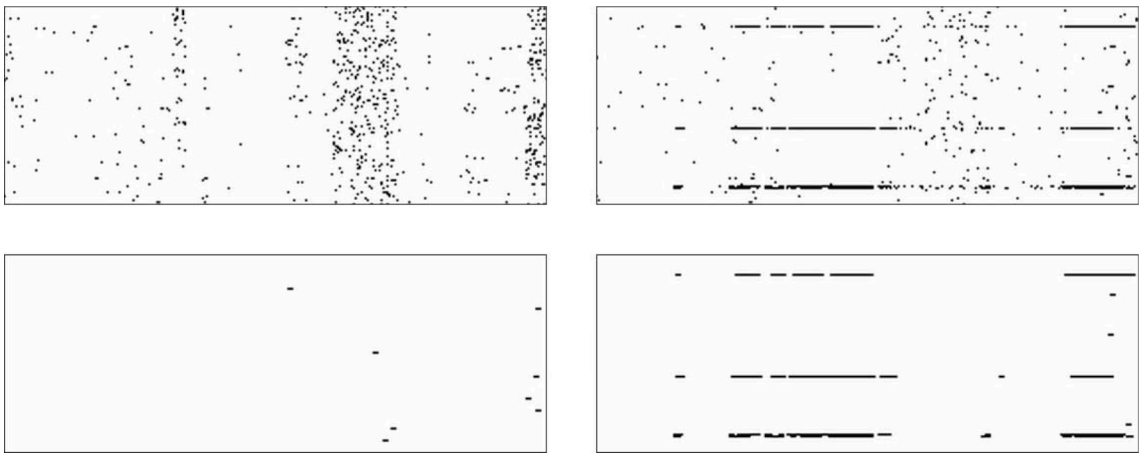


Abbildung 6.12: Merkmalsunterschiede vor (oben) und nach (unten) Anwendung eines zeitlichen Filters. Links nach verlustbehafteter Kompression und rechts nach Einfügen eines Objektes.

6.2.3 Gewährleistung der Sicherheit des Verfahrens

Um in sicherheitsrelevanten Szenarien Anwendung finden zu können muss die Sicherheit des Verfahrens gewährleistet sein. Einem Angreifer darf es nicht gelingen eine Fälschung durchzuführen, ohne entdeckt zu werden. Er darf also nicht wissen, wie

der Merkmalsvektor nach einer Manipulation aussieht. Um diesem Umstand Rechnung zu tragen haben wir in [TSS06] zwei Ansätze zur Gewährleistung der Sicherheit vorgestellt. Der erste Ansatz schlägt eine Gruppenbildung aufgrund pseudo-zufällig erzeugter Gruppen vor. Dies kann beispielsweise mittels einer Triangulation, die durch einen geheimen Schlüssel gesteuert wird, geschehen (siehe Abbildung 6.13). Durch den Schlüssel wird eine Triangulation erzeugt, durch die wiederum die Gruppen gebildet werden. Einem Angreifer ist es dann nicht möglich vorherzusagen, zu welcher Gruppe der Block gehört und ob er damit ein Maximum in seiner Gruppe darstellt. Zusätzliche Sicherheit bringt der zweite Ansatz aus [TSS06], bei dem nur die niedrigen Frequenzen zur Merkmalsgenerierung verwendet werden. Dadurch, dass die mittleren und hohen Frequenzen die Träger des Wasserzeichens sind und damit des authentifizierenden Merkmalsvektors, ist es für einen Angreifer schwierig nur Manipulationen durchzuführen, welche nur die niedrigen Frequenzen verändern. Schließlich sollte eine Manipulationen für einen Betrachter nicht offensichtlich sein. Eine Manipulation der mittleren und hohen Frequenzen würde vermutlich den Merkmalsvektor beschädigen und damit eine Manipulation anzeigen.

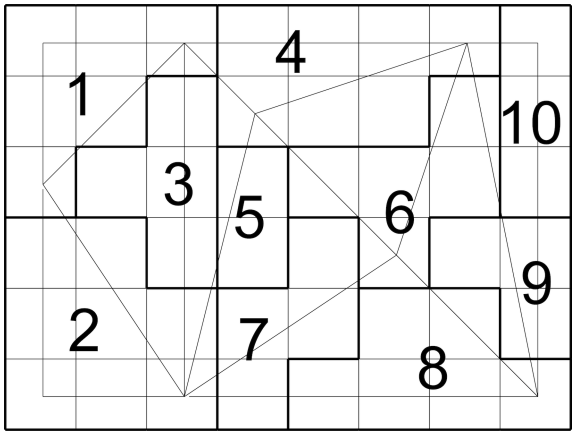


Abbildung 6.13: Bildung von Blockgruppen mittels Triangulation

Der Ansatz, dass Frame f_n durch sein nachfolgendes Frame f_{n+1} verifiziert wird, hat zur Folge, dass mindestens ein Frame des Videos nur indirekt authentifiziert werden kann. Um jedes Frame eines Videos verifizieren zu können, haben wir in [TSS05] die sogenannte „Verifikations-Kette“ eingeführt. Dazu werden alle Frames f_n in Gruppen F_q der Größe R unterteilt. Die gegenseitige Authentifizierung findet dann in der jeweiligen Gruppe statt. Abbildung 6.14 stellt ein Beispiel für eine Verifikations-Kette mit $R = 3$ dar. Die Gruppe F_q ist definiert als $F_q = \{f_n, f_{n+1}, f_{n+2}\}$. Neben dem Merkmalsvektor wird zusätzlich der Gruppenindex q eingebettet. Im Gegensatz zum eben vorgeschlagenen Verifikationsverfahren, verläuft die Verifikation in der Kette in entgegengesetzter Richtung. Jedes Frame der Gruppe enthält den Merkmalsvektor des ihm nachfolgenden Frames. In das letzte Frame wird der Merkmalsvektor des ersten Frames eingebettet. Würde die Verifikations-Kette in umgekehrter Richtung verlaufen, müsste Frame f_n bis zur Generierung von V_{n+2} im Speicher vorgehalten werden. So muss nur der deutlich kürzere Merkmalsvektor V_n zwischengespeichert werden. Durch die Anwendung der Verifikations-Kette können wir verschiedene Manipulationen der Zeitachse erkennen:

- Wird Frame f_{n+1} entfernt, so enthalten nur noch zwei Frames den Gruppenindex q. Da f_n weiterhin durch f_{n+2} verifiziert werden kann, f_{n+2} jedoch nicht verifiziert werden kann, können wir das Entfernen von f_{n+1} feststellen.
- Gleiches gilt für das Einfügen eines Frames f_{n+3}, dass in einem anderen Fall auch ein Frame ersetzen kann. Durch das Nicht-Vorhandensein von q kann diese Veränderung der Zeitachse erkannt werden.

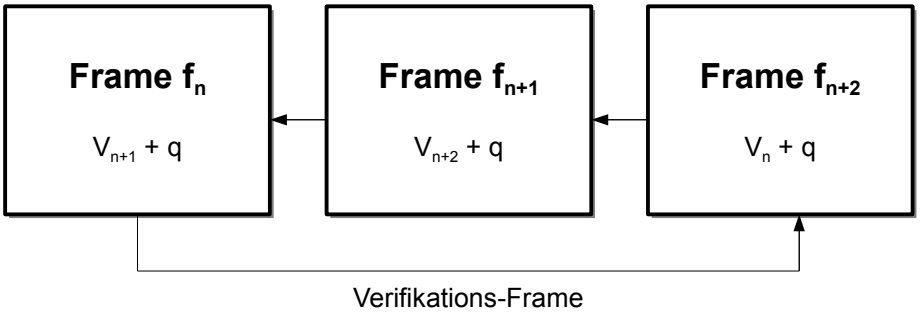


Abbildung 6.14: Verifikations-Kette

6.2.4 Analyse des Merkmals

Um das Merkmal zu analysieren, verwendeten wir das in Kapitel 5.1.5 vorgestellte Testvideo. Die untersuchten inhalts-erhaltenden Maßnahmen umfassten Formatumwandlung mit verlustbehafteter Kompression auf zwei verschiedene Bitraten und Formatumwandlung mit Skalierung auf drei verschiedene Auflösungen (siehe Abschnitt 6.1.3).

In der ersten Testreihe wurden unterschiedliche Blockgrößen (8×8 Pixel und 16×16 Pixel) und drei Gruppengrößen getestet (2, 4 bzw. 8 Blöcke pro Gruppe). Die Robustheits-Testergebnisse sind in Abbildung 6.15 dargestellt. Wiederum wurden für jede Block- und Gruppengröße der jeweils beste Parametersatz verwendet, d.h. welcher das beste Verhältnis zwischen Robustheit und Sensitivität darstellt. Die Parametersätze können Tabelle 6.4 entnommen werden. Aus Abbildung 6.15, welche die Total Rejection Rate (TRR) darstellt, können wir folgende Erkenntnisse ableiten:

- Mit größeren Blockgrößen steigt auch die Total Rejection Rate. Während bei einer Blockgröße von 8×8 Pixeln die durchschnittliche TRR bei 0,50% liegt, erhöht sie sich bei einer Blockgröße von 16×16 Pixeln auf 1,20%.
- Mit zunehmender Gruppengröße steigt auch die TRR. Liegt sie bei einer Gruppengröße von 2 Blöcken bei durchschnittlich 0,20% so steigt sie bei einer Gruppengröße von 8 Blöcken auf 1,31%.
- Die maximale Fehlerrate ist höher als die maximale TRR des Energiedifferenzverfahrens. Lag sie beim Energiedifferenzverfahren bei unter einem Prozent so

liegt sie bei einer Gruppengröße von 8 Blöcken und einer Blockgröße von 16×16 Pixeln, nach einer Skalierung auf 50% der Originalauflösung, bei 3,09%.

In den Abbildungen 6.16 und 6.17 sind die Correct Rejection Rate (CRR) und die False Rejection Rate (FRR) für die oben genannten Parametersätze dargestellt. Die inhalts-verändernden Maßnahmen umfassten das Austauschen von Blockpaaren und das Einfügen bzw. Entfernen von Einzelblöcken (siehe Abschnitt 6.1.3). Aus den Ergebnissen können wir folgende Erkenntnisse ableiten:

- Mit größerer Blockgröße sinkt die CRR. Während bei einer Blockgröße von 8×8 Pixeln die durchschnittliche CRR bei 87,89% liegt, sinkt sie bei einer Blockgröße von 16×16 Pixeln auf 62,23%.
- Eine größere Gruppengröße wirkt sich positiv auf die CRR aus. Bei einer Blockgröße von 8×8 Pixeln steigt die durchschnittliche CRR von 80,73% (2 Blöcke pro Gruppe) auf bis zu 92,30% (8 Blöcke pro Gruppe). Ähnliches kann bei einer Blockgröße von 16×16 Pixeln beobachtet werden, wobei sich hier die Gruppengröße mit 4 Blöcken als die mit der durchschnittlich besten CRR herausgestellt hat.
- Die Ergebnisse sind deutlich besser als für das Energiedifferenzverfahren. Erzielten wir mit dem Energiedifferenzverfahren maximal eine CRR von 45,94%, so liegt die maximale CRR beim Entropie-Verfahren bei 92,30%.
- Bei einer Blockgröße von 8×8 Pixeln ist die Differenz zwischen der CRR bei 5% Veränderung (83,37%) ähnlich zu der CRR bei 10% (92,40%) Veränderung.

In einem weiteren Schritt analysierten wir die Einführung des Quantisierungsfaktors QF auf die Robustheit und Sensitivität des Merkmals. Aus Übersichtsgründen wurde nur eine Blockgröße von 8×8 Pixeln verwendet. Aus Tabelle 6.5 sind die verwendeten Parametersätze zu entnehmen. Die Testergebnisse sind in den Abbildungen 6.18, 6.19 und 6.20 dargestellt. Aus den Ergebnissen können folgende Erkenntnisse abgeleitet werden:

- Mit einem höheren Quantisierungsfaktor nimmt die Robustheit des Verfahrens zu. Die durchschnittliche TRR sinkt von 2,58% bei $QF = 1$ auf bis zu 0,60% bei $QF = 50$. Die maximale TRR bei einer Skalierung auf 50% der Originalauflösung sinkt dabei auf 1,35%.
- Ein höherer Quantisierungsfaktor wirkt sich nicht negativ auf die Sensitivität des Verfahrens aus. Bei einem Quantisierungsfaktor $QF = 10$ erzielten wir das Maximum mit einer durchschnittlichen CRR von 92,94%. Ähnlich ist das Ergebnis bei $QF = 25$ mit einer durchschnittlichen CRR von 92,30%.

In einem dritten Schritt analysierten wir die Einführung des zeitlichen Filters auf die Robustheit und Sensitivität des Merkmals. Aus Übersichtsgründen wurde wiederum

nur eine Blockgröße von 8×8 Pixeln verwendet. Aus Tabelle 6.6 sind die verwendeten Parametersätze zu entnehmen. Aus den Testergebnissen in den Abbildungen 6.21, 6.22 und 6.23 können wir folgende Erkenntnisse ableiten:

- Mit steigender Filterlänge steigt auch die Robustheit des Verfahrens. Die durchschnittliche TRR sinkt von 2,32% (Filterlänge 1s) auf bis zu 0,20% (Filterlänge 5s).
- Ein längerer Filter wirkt sich nicht negativ auf die Sensitivität des Verfahrens aus. Wir erzielten die beste durchschnittliche CRR bei einer Filterlänge von 3s (92,30%) bzw. 2s (92,03%).

Fazit: Das Verfahren ist gut geeignet, um Merkmalsvektoren zu generieren, die in Integritäts-Videowasserzeichen zum Einsatz kommen können.

Blockgröße	Gruppengröße	Quantisierungsfaktor QF	Filterlänge
8×8	2	25	3s
8×8	4	10	3s
8×8	8	25	3s
16×16	2	1	2s
16×16	4	1	2s
16×16	8	10	2s

Tabelle 6.4: Parametersätze für die verschiedenen Block- und Gruppengrößen

Quantisierungsfaktor QF	Gruppengröße	Filterlänge
1	4	3s
10	8	3s
25	8	3s
50	8	3s

Tabelle 6.5: Parametersätze für die Werte von QF

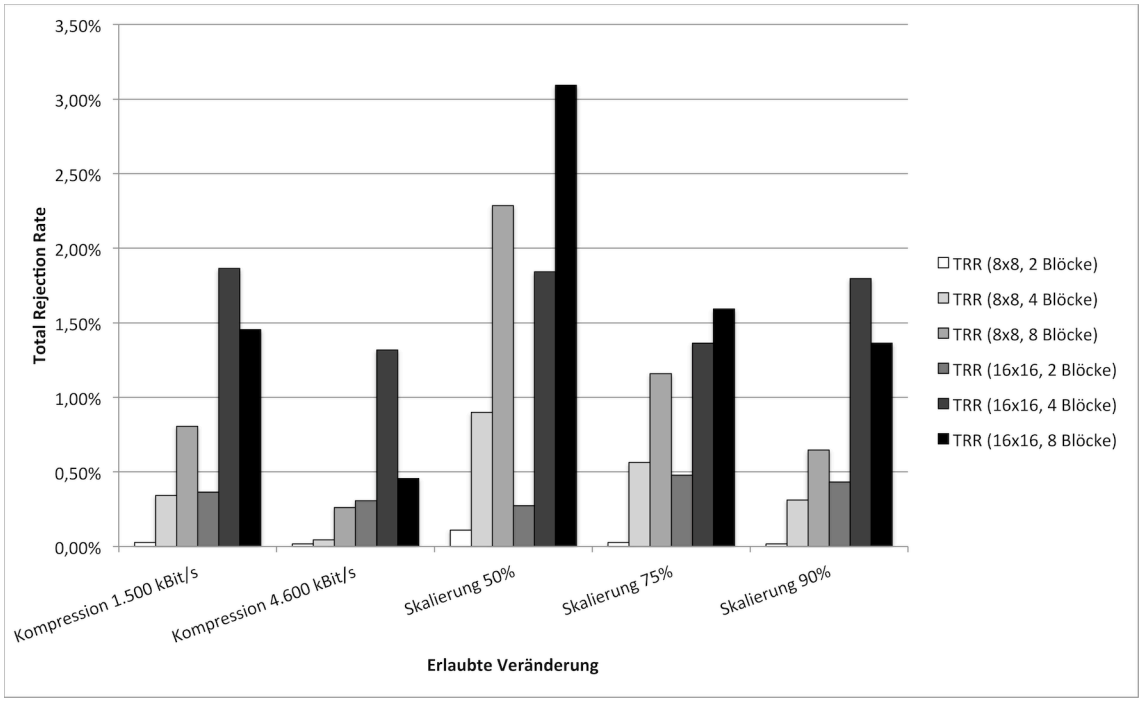


Abbildung 6.15: TRR des Entropie-Verfahrens gruppiert nach Block- und Gruppengröße

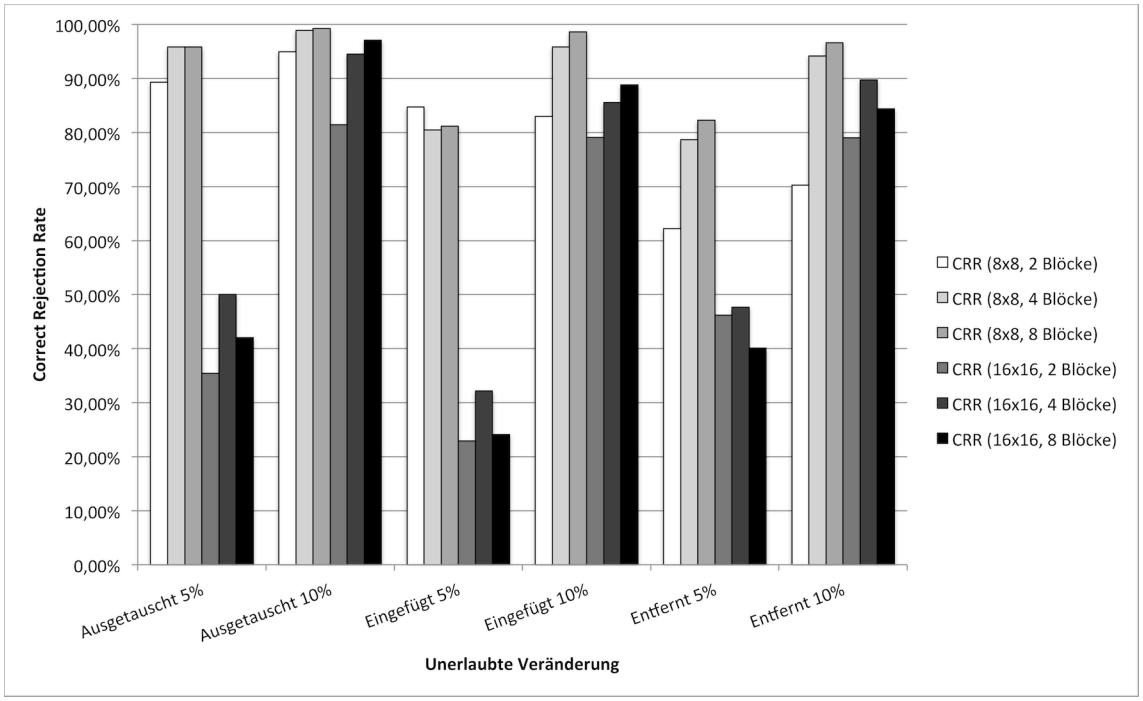


Abbildung 6.16: CRR des Entropie-Verfahrens gruppiert nach Block- und Gruppengröße

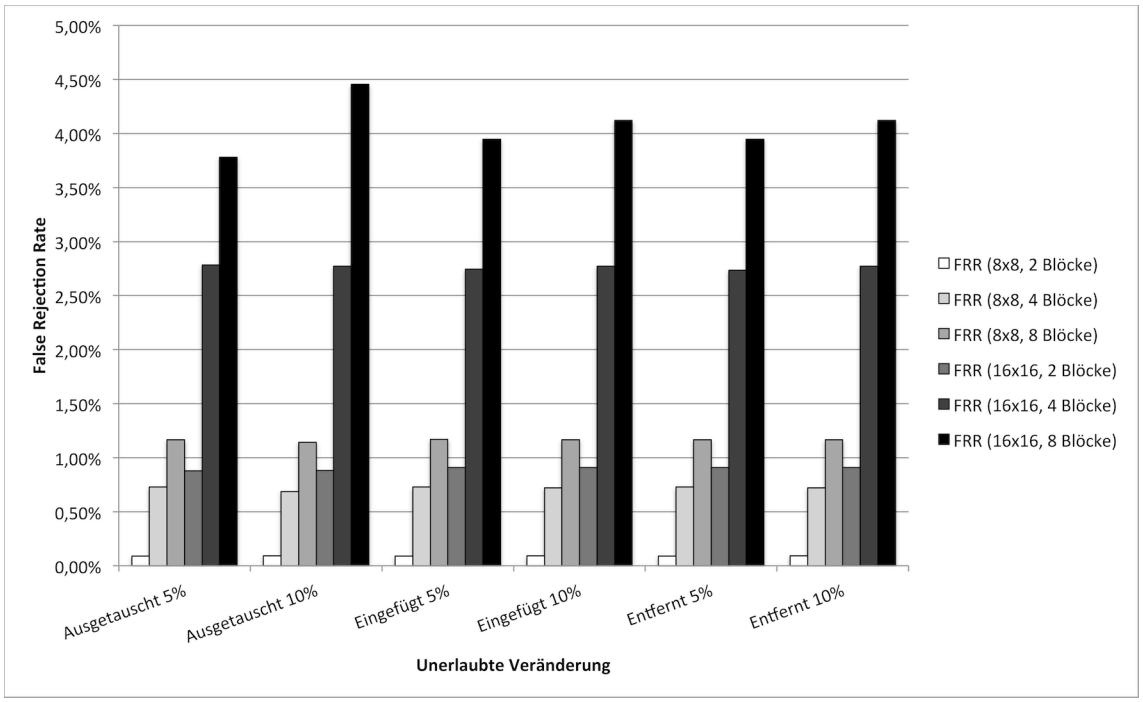


Abbildung 6.17: FRR des Entropie-Verfahrens gruppiert nach Block- und Gruppengröße

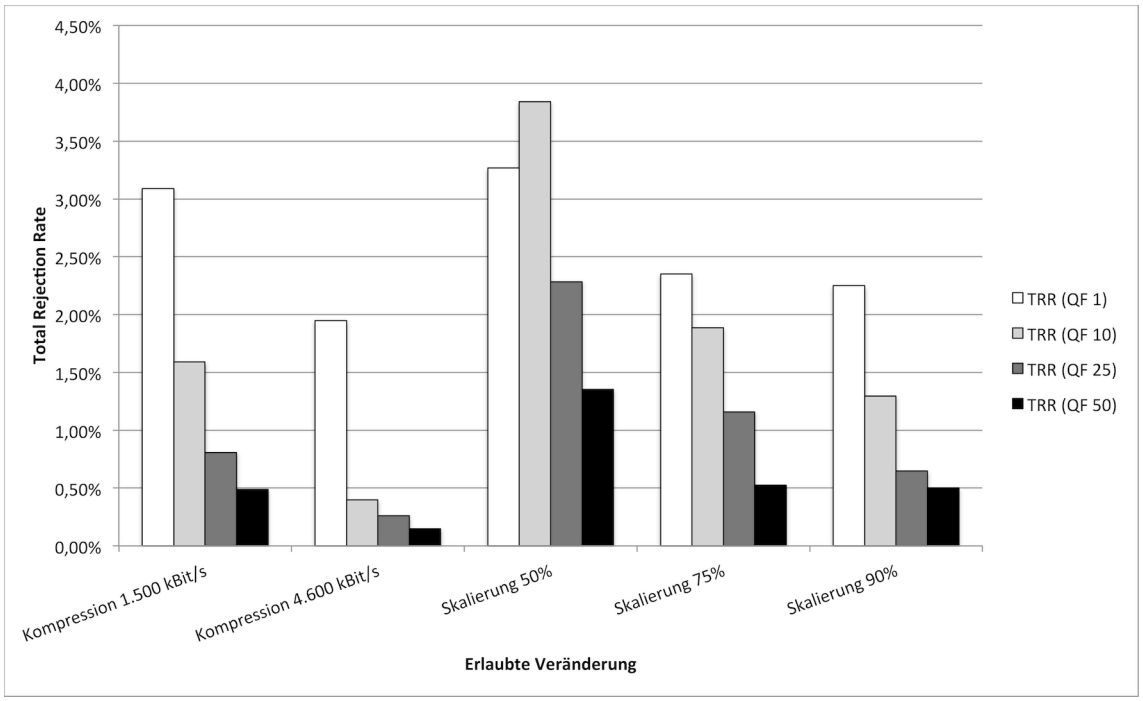


Abbildung 6.18: TRR des Entropie-Verfahrens gruppiert nach Quantisierungsfaktor QF

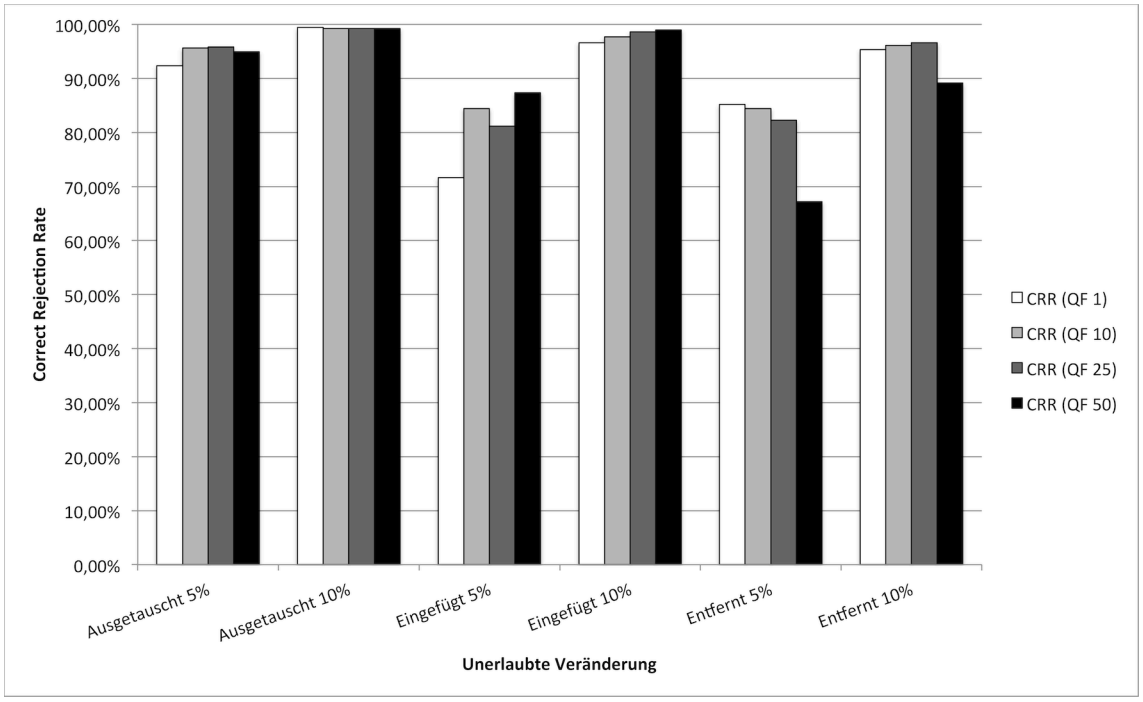


Abbildung 6.19: CRR des Entropie-Verfahrens gruppiert nach Quantisierungsfaktor QF

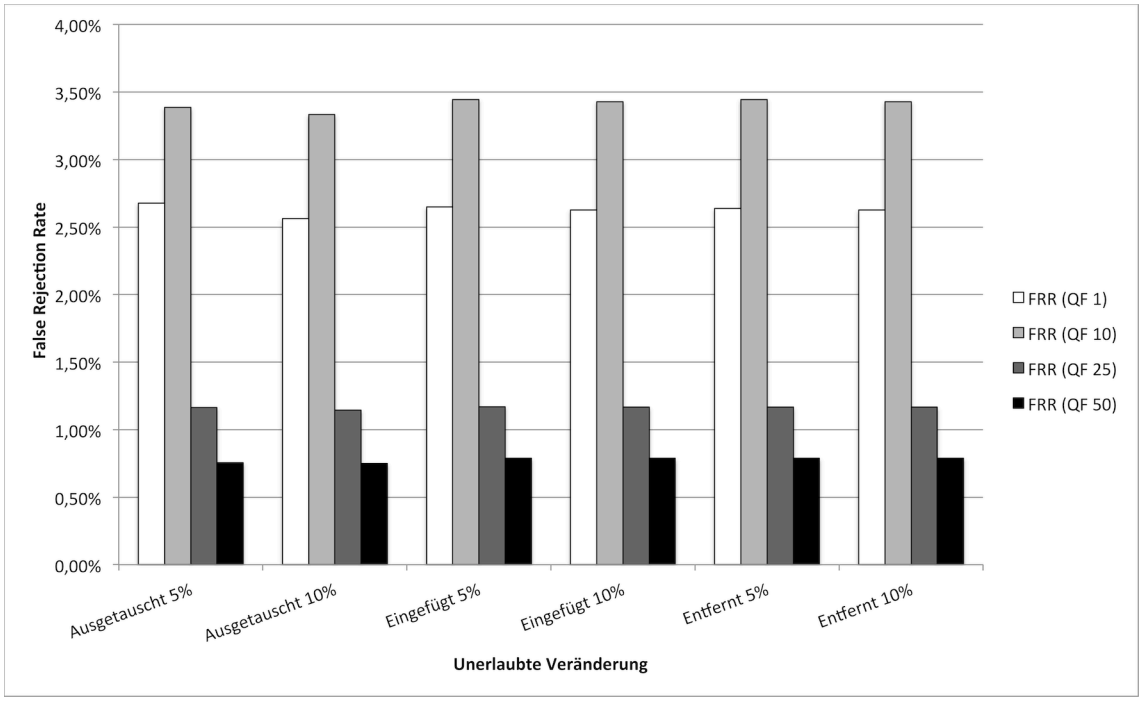


Abbildung 6.20: FRR des Entropie-Verfahrens gruppiert nach Quantisierungsfaktor QF

Filterlänge	Gruppengröße	Quantisierungsfaktor QF
1s	2	25
2s	4	25
3s	8	25
4s	8	10
5s	8	10

Tabelle 6.6: Parametersätze für die Filterlängen

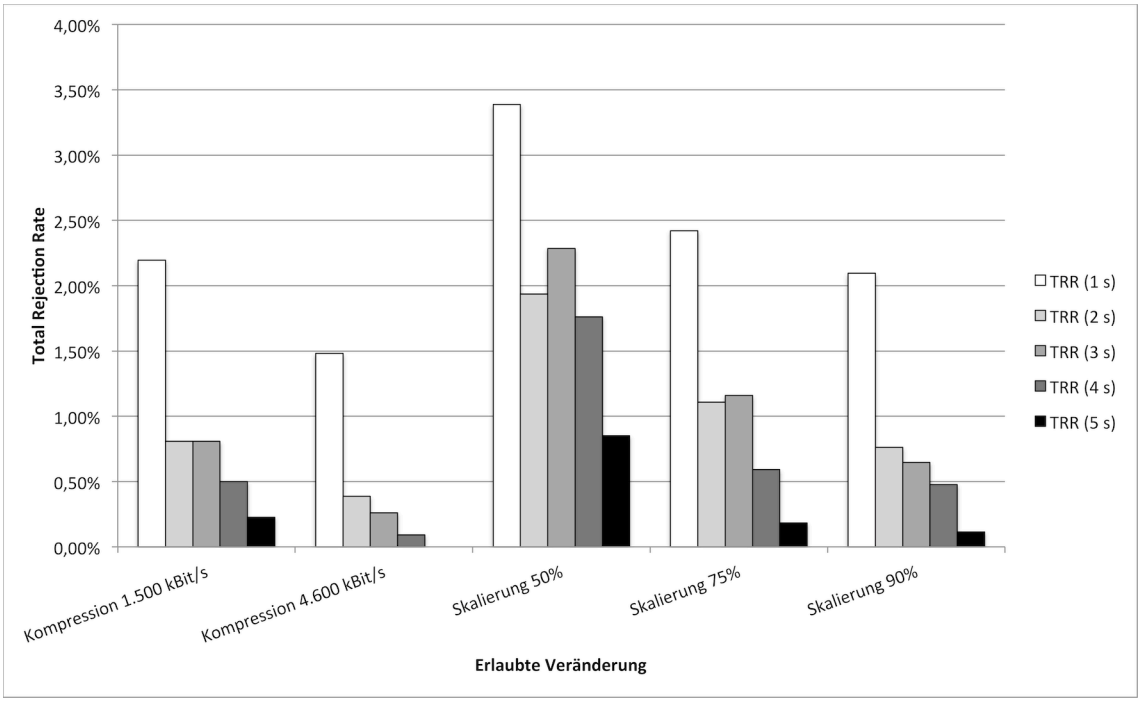


Abbildung 6.21: TRR des Entropie-Verfahrens gruppiert nach Filterlänge

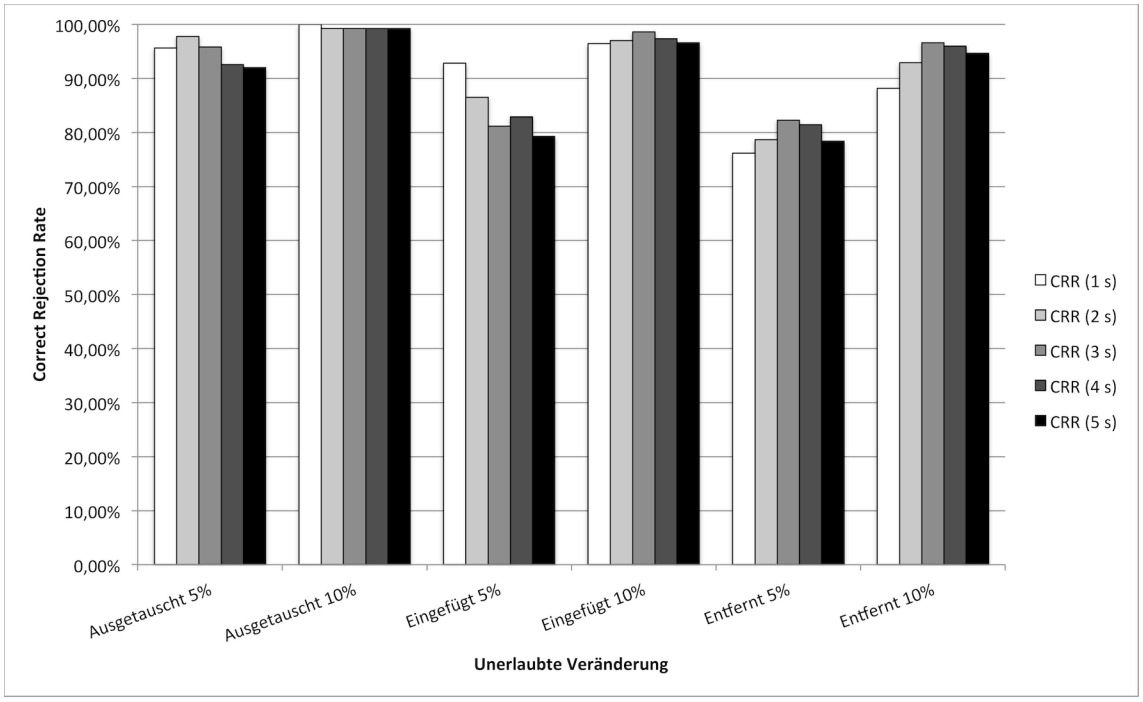


Abbildung 6.22: CRR des Entropie-Verfahrens gruppiert nach Filterlänge

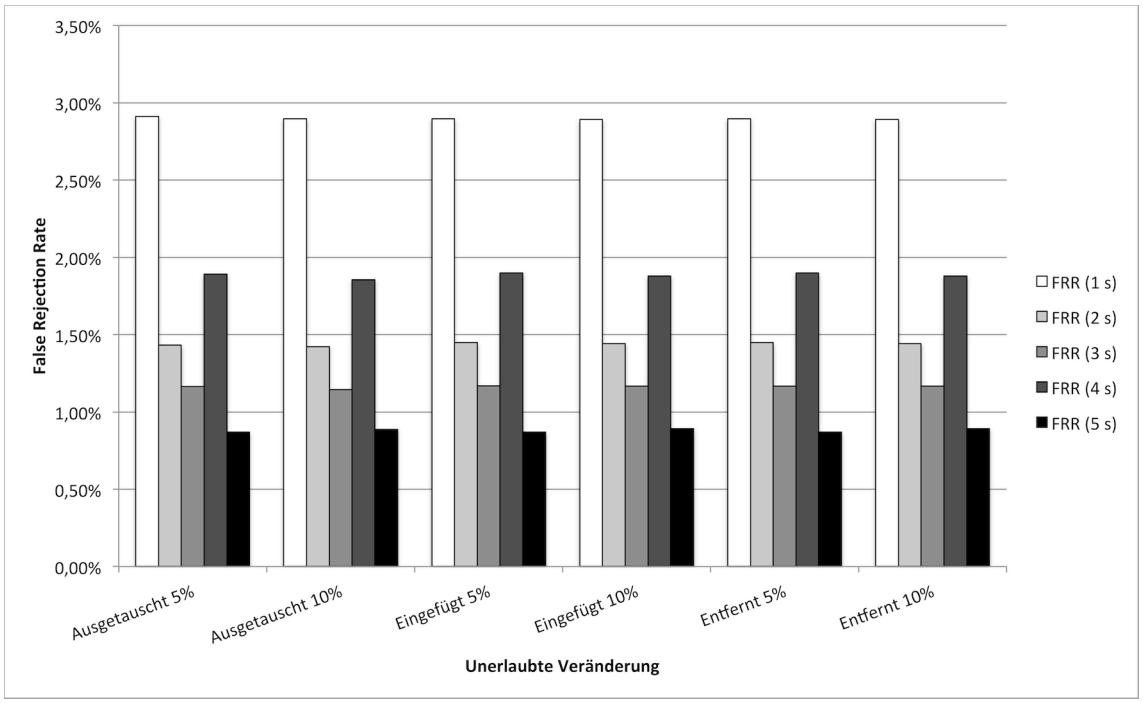


Abbildung 6.23: FRR des Entropie-Verfahrens gruppiert nach Filterlänge

6.3 Interest-Operator nach Moravec

Neben der Abbildung von nicht-inhaltlichen Abhängigkeiten auf Merkmalsvektoren, untersuchen wir die Abbildung inhaltlicher Abhängigkeiten. Eine Möglichkeit inhaltliche Abhängigkeiten auf Merkmalsvektoren abzubilden, ist die Verwendung so genannter Interest-Operatoren. Sie haben ihren Ursprung in der Robotik, wobei sie Robotern markante (interessante) Punkte im Raum zurückliefern sollen. Die gefundenen Punkte sollen so dominant sein, dass sie in einer Sequenz von aufeinander folgenden Frames zu finden sind. Anhand dieser Punkte bestimmen die Roboter ihre eigene Position und können sich damit in Räumen orientieren. Erste Vertreter dieser Interest-Operatoren wurden von Moravec [Mor77] im Jahr 1977 und von Förstner/Gülch [FG87] im Jahr 1987 vorgestellt.

Aufgrund seiner geringen Komplexität bei gleichzeitig guter Trennschäfe untersuchen wir in diesem Abschnitt den Interest-Operator von Moravec auf seine Tauglichkeit für den Einsatz zur Generierung von Merkmalsvektoren. Nachfolgend beschreiben wir die Arbeitsweise des ursprünglichen Interest-Operators und stellen anschließend unsere Modifikationen vor, um den Operator für den Einsatz in Videowasserzeichen zum Integritätsschutz zu optimieren.

Der ursprüngliche Interest-Operator von Moravec basiert auf Grauwerten. Er berechnet für ein Pixel die Abweichung zu seinen Nachbarpixeln in horizontaler, vertikaler und zwei diagonalen Richtungen.

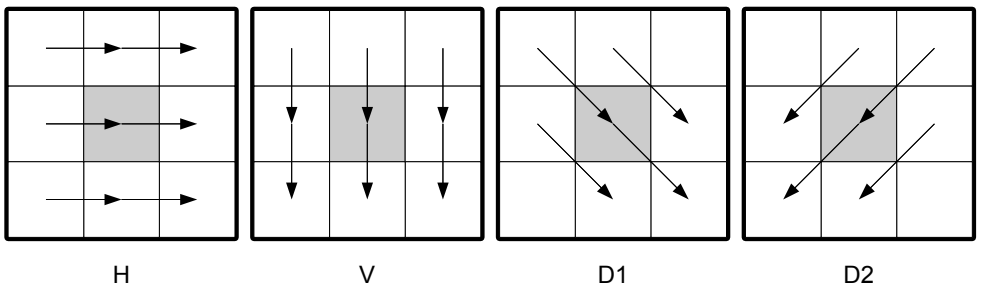


Abbildung 6.24: Hauptrichtungen des Interest-Operators nach Moravec (H = Horizontal, V = Vertikal, D1 = Diagonale 1, D2 = Diagonale 2)

Sei $p_{x,y}$ der Grauwert eines Pixels an der Position (x, y) und $n \times n$ die Größe eines Fensters, der umliegenden Pixelwerte mit $p_{x,y}$ im Zentrum. Dann berechnen sich die Werte $H(x, y)$ (horizontal), $V(x, y)$ (vertikal), $D1(x, y)$ (Diagonale 1) und $D2(x, y)$ (Diagonale 2) der vier Richtungen wie folgt [Sch99]:

$$V(x,y) = \sum_{r,s} |p_{x+r,y+s} - p_{x+r,y+s+1}| \tag{6.7}$$

$$H(x,y) = \sum_{r,s} |p_{x+r,y+s} - p_{x+r+1,y+s}| \tag{6.8}$$

$$D1(x,y) = \sum_{r,s} |p_{x+r,y+s} - p_{x+r+1,y+s+1}| \tag{6.9}$$

$$D2(x,y) = \sum_{r,s} |p_{x+r,y+s} - p_{x+r+1,y+s-1}| \tag{6.10}$$

Dabei gilt $-\frac{n}{2} \leq r, s \leq \frac{n}{2}$.

Sei $F = \{f_1, \ldots, f_N\}$ die Menge aller Videoframes in einem Video und $B_n = \{b_{n,1}, \ldots, b_{n,M}\}$ die Menge aller Blöcke in Frame $f_n \in F$. Dann berechnet sich der Interestwert $I(n,m,x,y)$ eines Pixels an Position (x,y) in Block $b_{n,m}$ wie folgt:

$$I(n,m,x,y) = min(H(x,y), V(x,y), D1(x,y), D2(x,y)) \tag{6.11}$$

Das Minimum stellt sicher, dass nur allein stehende, signifikante Punkte erkannt werden. Punkte auf Ecken und Kanten werden von Moravecs Interest-Operator als nicht markant angesehen. Dadurch wird das Ergebnis robuster gegenüber Rauschen und inhalts-erhaltenden Veränderungen.

6.3.1 Generierung des Merkmalsvektors und Einbettung

Analog zum Verfahren aus Abschnitt 6.2 wird zunächst das Frame f_n auf eine feste Größe skaliert (hier 128×128 Pixel) um eine feste Länge des Merkmalsvektors unabhängig von der Frameauflösung zu erreichen. Danach wird jeder Block $b_{n,m}$ wie folgt vorverarbeitet:

1. Anwendung eines Tief-Pass-Filters auf den Block. Damit werden die mittleren und hohen Frequenzen von der Merkmalsgenerierung ausgeschlossen. Diese Frequenzen werden im späteren Einbettungsprozess verwendet.
2. Anwendung eines Weichzeichen-Filters, um das gefilterte Bild zu glätten.
3. Berechnung des Block-Interestwertes $I(b_{n,m})$, indem die Summe aller Pixel-Interestwerte in $b_{n,m}$ gebildet wird.

Wir unterteilen die Blöcke in Gruppen $G_n = \{G_{n,1}, \ldots, G_{n,O}\}$. Ein Vektorbit des Merkmalsvektors $V_n = v_{n,1} \| \ldots \| v_{n,M}$ des Frames f_n wird nach [TSS05] wie folgt gebildet:

- Ist $I(b_{n,m})$ innerhalb seiner Gruppe $G_{n,o}$ ein Maximum, dann wird sein zugehöriges Vektorbit $v_{n,m}$ auf 1 gesetzt.
- Ist $I(b_{n,m})$ innerhalb seiner Gruppe $G_{n,o}$ größer als die Hälfte aller $I(b_{n,m})$ in $G_{n,o}$, dann wird sein zugehöriges Vektorbit $v_{n,m}$ auf 1 gesetzt.

- Erfüllt $I(b_{n,m})$ keine der vorherigen Bedingungen, so wird sein zugehöriges Vektorbit $v_{n,m}$ auf 0 gesetzt.

Der Merkmalsvektor V_n wird mit einem robusten Wasserzeichenverfahren in das Frame f_{n+1} eingebettet. Dabei ist wie im vorherigen Abschnitt zu beachten, dass die Einstellungen so vorgenommen werden, so dass die niedrigen Frequenzen, die zur Berechnung des Merkmalsvektors verwendet werden, nicht durch den Einbettungsprozess verändert werden. Zusätzlich wird ein eindeutiger, kontinuierlich steigender Frameindex dem Merkmalsvektor angehängt, um Manipulationen an der Zeitachse zu erkennen.

Abbildung 6.25 stellt das Verfahren zur Generierung des Merkmalsvektors noch einmal schematisch dar. Das Frame f_n wird zunächst vorverarbeitet und dann werden die Interest-Werte berechnet. Anschließend werden Gruppen gebildet und der Merkmalsvektor berechnet. Vektor V_n wird dann unter Kontrolle eines geheimen Schlüssels K in das Frame f_{n+1} eingebettet.

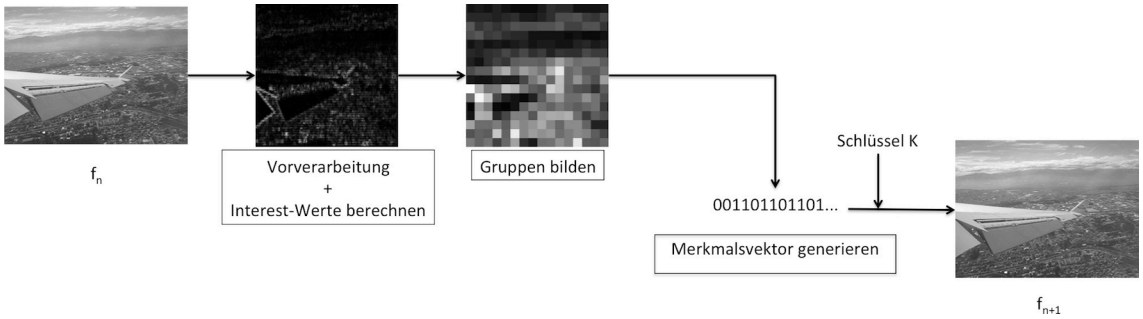


Abbildung 6.25: Merkmalsgenerierung für das Verfahren mit dem Moravec-Interest-Operator

Abbildung 6.26 stellt ein Frame beispielhaft seinen Block-Interestwerten gegenüber. Auf der linken Seite ist das Frame abgebildet mit seinen Interest-Werten in der Mitte, die auf Basis des Moravec-Interest-Operators berechnet wurden. Auf der rechten Seite sind die Interest-Werte pro Block abgebildet. Sie wurden zur besseren Sichtbarkeit gespreizt. Ein dunklerer Block verfügt also über weniger Interest-Werte. Es ist zu erkennen, dass sich die meisten Interest-Werte im unteren und mittleren Bereich befinden. Der Himmel wird vom Moravec-Interest-Operator als weniger signifikant erkannt.

6.3.2 Auslesen des Merkmalsvektors und Verifikation

Im Ausleseprozess wird der Merkmalsvektor V_n aus dem Frame f_{n+1} ausgelesen. Wir berechnen den aktuellen und möglicherweise modifizierten Merkmalsvektor $\tilde{V}_n$ aus Frames f_n. Unterscheiden sich V_n und $\tilde{V}_n$, so ist von einer Manipulation des Videos auszugehen. Anhand der in Abschnitt 6.2.2 vorgestellten Entscheidungshilfen kann analysiert werden, an welchen Positionen das Video manipuliert wurde.

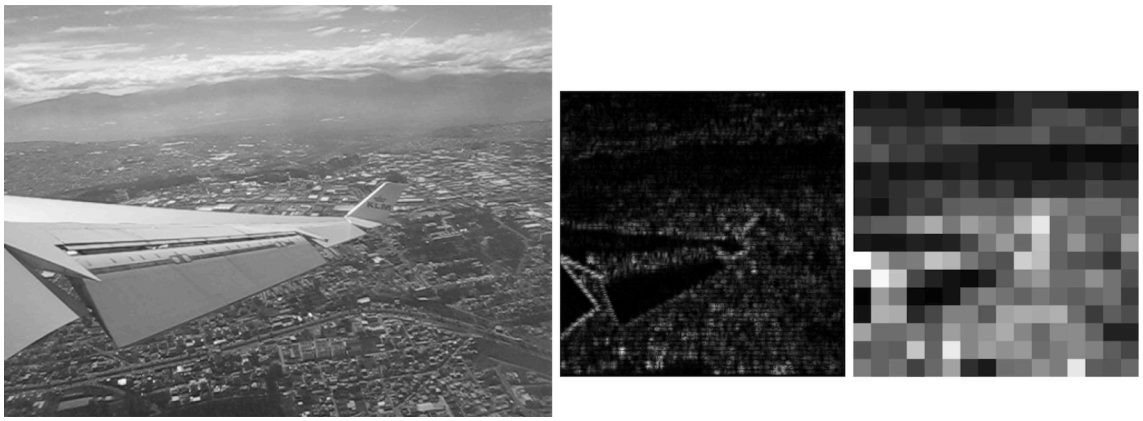


Abbildung 6.26: Gegenüberstellung eines Frames mit seinen Block-Interestwerten nach Moravec

Um inhalts-erhaltende Maßnahmen von inhalts-verändernden Maßnahmen zu unterscheiden, verwenden wir den in Abschnitt 6.2.2 eingeführten zeitlichen Filter.

6.3.3 Erweiterung des Verfahrens

In [TTS08] stellen wir eine Erweiterung des auf Helligkeitswerten basierenden Verfahrens vor, die von Schimmel in [Sch99] erstmalig beschrieben wird. Ziel ist zum einen eine bessere Trennschärfe zu erreichen und zum anderen das Erkennen von Farbmanipulationen zu ermöglichen. Farbmanipulationen sollten erkannt werden können, da zwei unterschiedliche Farben über die gleiche Helligkeit, d.h. den gleichen Grauwert verfügen können. Diese Problematik wurde in Kapitel 4.3 angesprochen. Dort haben wir gezeigt, dass die Einfärbung einer Wasserlache die Bildaussage veränderte.

Um eine bessere Trennschärfe zu erreichen werden vier zusätzliche Hauptrichtungen eingeführt. Dadurch können auch Grauwertübergänge in diesen Richtungen analysiert und allein stehende Punkte besser erkannt werden. Abbildung 6.27 zeigt die neuen diagonalen Hauptrichtungen.

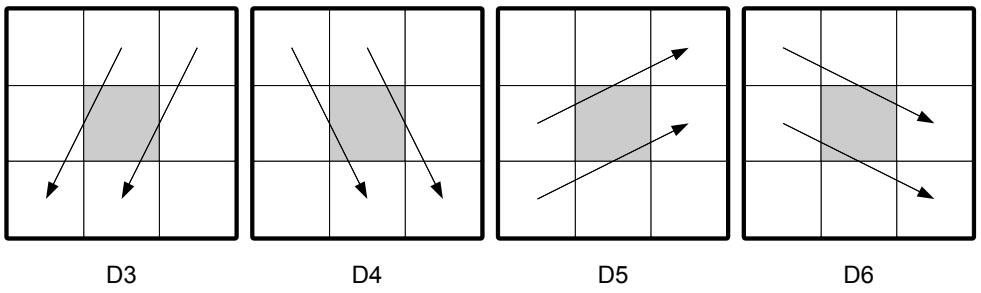


Abbildung 6.27: Zusätzliche Hauptrichtungen des Interest-Operators

Die Formeln der zusätzlichen Hauptrichtungen sind wie folgt [Sch99]:

$$D3(x,y) = \sum_{r,s} |p_{x+r,y+s+1} - p_{x+r+1,y+s-1}| \tag{6.12}$$

$$D4(x,y) = \sum_{r,s} |p_{x+r-1,y+s} - p_{x+r+1,y+s-1}| \tag{6.13}$$

$$D5(x,y) = \sum_{r,s} |p_{x+r-1,y+s} - p_{x+r+1,y+s+1}| \tag{6.14}$$

$$D6(x,y) = \sum_{r,s} |p_{x+r,y+s-1} - p_{x+r+1,y+s+11}| \tag{6.15}$$

Wiederum gilt $-\frac{n}{2} \leq r, s \leq \frac{n}{2}$.

Der Interestwert $I(n,m,x,y)$ für ein Pixel $p_{x,y}$ in Block $b_{n,m}$ wird dann aus dem Minimum aller acht Hauptrichtungen berechnet.

Um auch Manipulationen an den Farben erkennen zu können, wird der Interest-Operator wie von Schimmel beschrieben [Sch99] von Grau- auf Farbwerte erweitert. Dazu wird der Interestwert eines Pixels nicht aus Grauwert-Differenzen sondern aus Farbwertdifferenzen gebildet. Schimmel verwendet statt der euklidischen Abstandsnorm die Maximums-Norm, um die unterschiedliche quantitative Bewertung von Farbübergängen zu umgehen.

Seien r_{px}, g_{px}, b_{px} die Rot-, Grün- und Blaukanalwerte eines Pixels an Position px im RGB-Farbraum. Dann berechnet sich die Pixeldifferenz zweier Pixel an den Positionen px_1 und px_2 wie folgt:

$$max(|r_{px_1} - r_{px_2}|, |g_{px_1} - g_{px_2}|, |b_{px_1} - b_{px_2}|) \tag{6.16}$$

Die Grauwertdifferenz zwischen zwei Pixeln wird also durch das Maximum der drei Farbkanaldifferenzen ersetzt.

6.3.4 Analyse des Merkmals

Für die Analyse des Merkmals wurde wieder das in Kapitel 5.1.5 vorgestellte Testvideo verwendet. Die getesteten inhalts-erhaltenden Maßnahmen umfassten Formatumwandlung mit verlustbehafteter Kompression und Formatumwandlung mit Skalierung. In einer ersten Testreihe untersuchten wir zwei verschiedene Blockauflösungen (8×8 und 16×16) und drei verschiedene Gruppengrößen (2, 4 und 8 Blöcke). Die jeweils besten Parametersätze mit ausgewogener Robustheit und Sensitivität können der Tabelle 6.7 entnommen werden.

Abbildung 6.28 enthält die Robustheitsergebnisse für die Blockauflösungen und Gruppengrößen. Aus den Ergebnissen können folgende Erkenntnisse abgeleitet werden:

Blockgröße	Gruppengröße	Filterlänge
8×8	2	2s
8×8	4	2s
8×8	8	3s
16×16	2	2s
16×16	4	2s
16×16	8	3s

Tabelle 6.7: Parametersätze für die verschiedenen Block- und Gruppengrößen

- Mit zunehmender Gruppengröße steigt auch die Total Rejection Rate (TRR). Bei einer Gruppengröße von 2 Blöcken erreichten wir eine TRR von durchschnittlich 1,25%. Sie steigt bei einer Gruppengröße von 8 Blöcken auf bis zu 4,67%.
- Eine kleinere Blockgröße zieht eine höhere TRR nach sich. Bei einer Auflösung von 8×8 Pixeln beträgt sie 4,33% während sie bei einer Auflösung von 16×16 Pixeln auf 1,82% absinkt. Sie verhält sich hier also genau umgekehrt zum Entropie-Verfahren.

In den Abbildungen 6.29 und 6.30 sind die Ergebnisse der inhalts-verändernden Maßnahmen dargestellt. Sie beinhalteten das Einfügen und Entfernen von Einzelblöcken unterschiedlicher Größe, sowie den Austausch von Blockpaaren unterschiedlicher Größe. Folgendes kann man beobachten:

- Mit zunehmender Blockgröße sinkt die Correct Rejection Rate (CRR). Sie beträgt durchschnittlich 86,38% bei einer Blockgröße von 8×8 Pixeln und 53,30% bei 16×16 Pixeln.
- Die Gruppengröße hat keinen einheitlichen Einfluss auf die CRR. Allerdings ist eine mittlere Gruppengröße die mit der durchschnittlich höchsten False Rejection Rate (FRR). Diese schwankt zwischen 3,80% (bei 16×16 Pixeln) und 5,80% (bei 8×8 Pixeln).

In der zweiten Testreihe untersuchten wir den Einfluss der zusätzlichen Diagonalen auf die Qualität des Merkmals. Die Abbildungen 6.31, 6.32 und 6.33 zeigen die Testergebnisse für eine Blockgröße von 8×8 Pixeln und eine Gruppengröße von 4 Blöcken. Für diese Testreihe wurden die Parametersätze aus Tabelle 6.7 verwendet. Wir konnten folgendes feststellen:

- Die Einführung zusätzlicher Diagonalen hatte keinen signifikanten Einfluss auf die TRR. Sie ist mit 4,94% ähnlich zu der durchschnittlichen TRR des Verfahrens ohne zusätzliche Diagonalen (4,72%).

- Auch bei der Sensitivität konnte keine signifikante Verbesserung erzielt werden. Die CRR ist mit 91,41% annähernd gleich geblieben (vorher 90,42%).

In der dritten Testreihe wurde die Einführung der Farbwerte untersucht. In den Abbildungen 6.34, 6.35 und 6.36 sind die Testergebnisse für eine Blockgröße von 8×8 Pixeln, eine Gruppengröße von 4 Blöcken und mit zusätzlichen Diagonalen. Folgendes kann beobachtet werden:

- Die Verwendung von Farbwerten führte nur zu einem minimalen Abfall der durchschnittlichen CRR. Sie beträgt jetzt 90,88% (vorher 91,41%).
- Mit der Verwendung von Farbwerten konnte eine Verbesserung der Robustheit erzielt werden. Die durchschnittliche TRR beträgt jetzt 3,44% (vorher 4,94%).

Fazit: Das Moravec-Verfahren zeichnet sich ähnlich wie das Entropie-Verfahren mit einer guten Sensitivität gegenüber inhalts-verändernden Maßnahmen aus. Allerdings ist es weniger robust, als das Entropie-Verfahren.

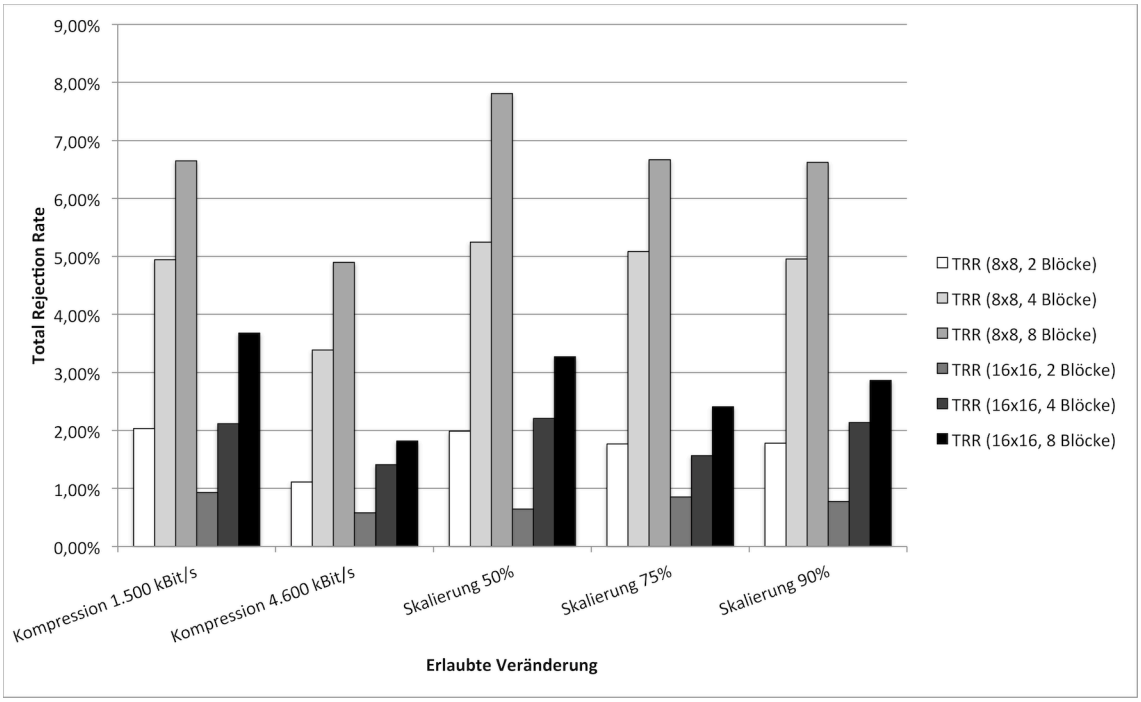


Abbildung 6.28: TRR des Moravec-Verfahrens gruppiert nach Block- und Gruppengröße

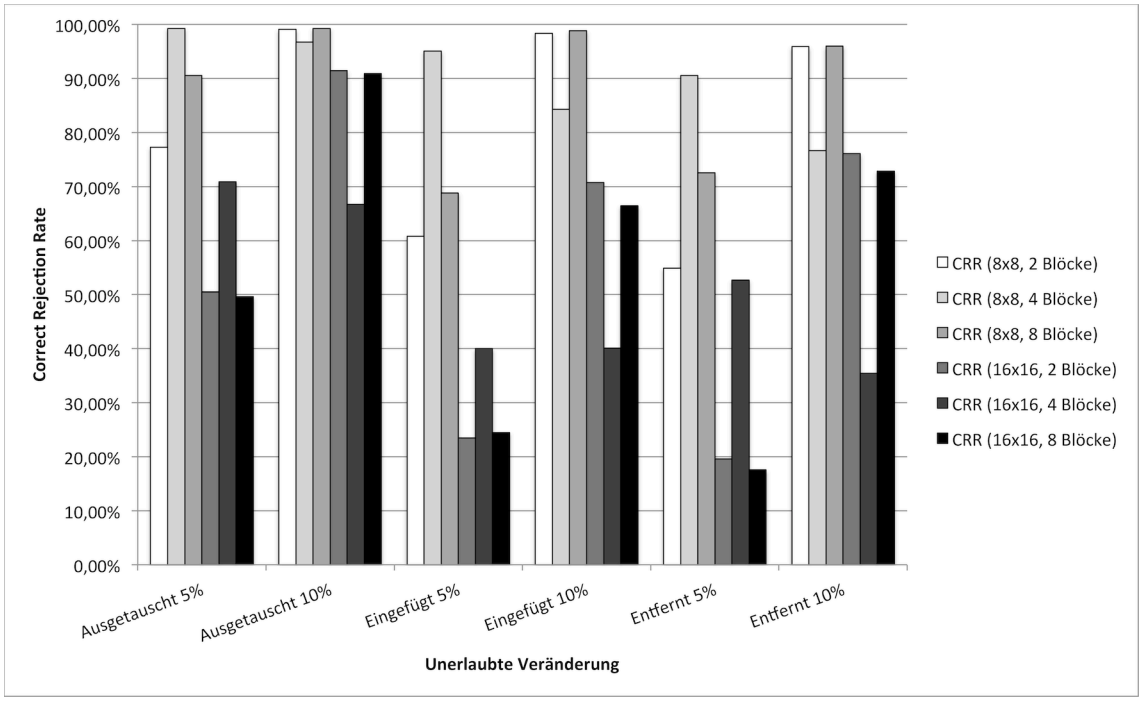


Abbildung 6.29: CRR des Moravec-Verfahrens gruppiert nach Block- und Gruppengröße

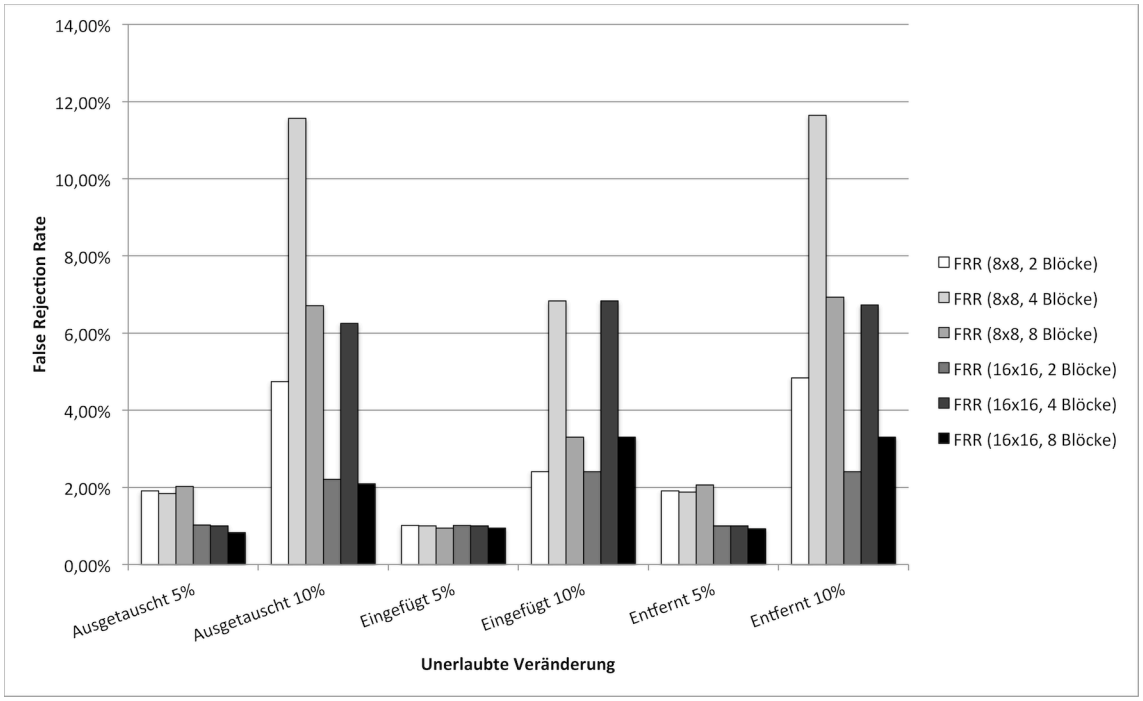


Abbildung 6.30: FRR des Moravec-Verfahrens gruppiert nach Block- und Gruppengröße

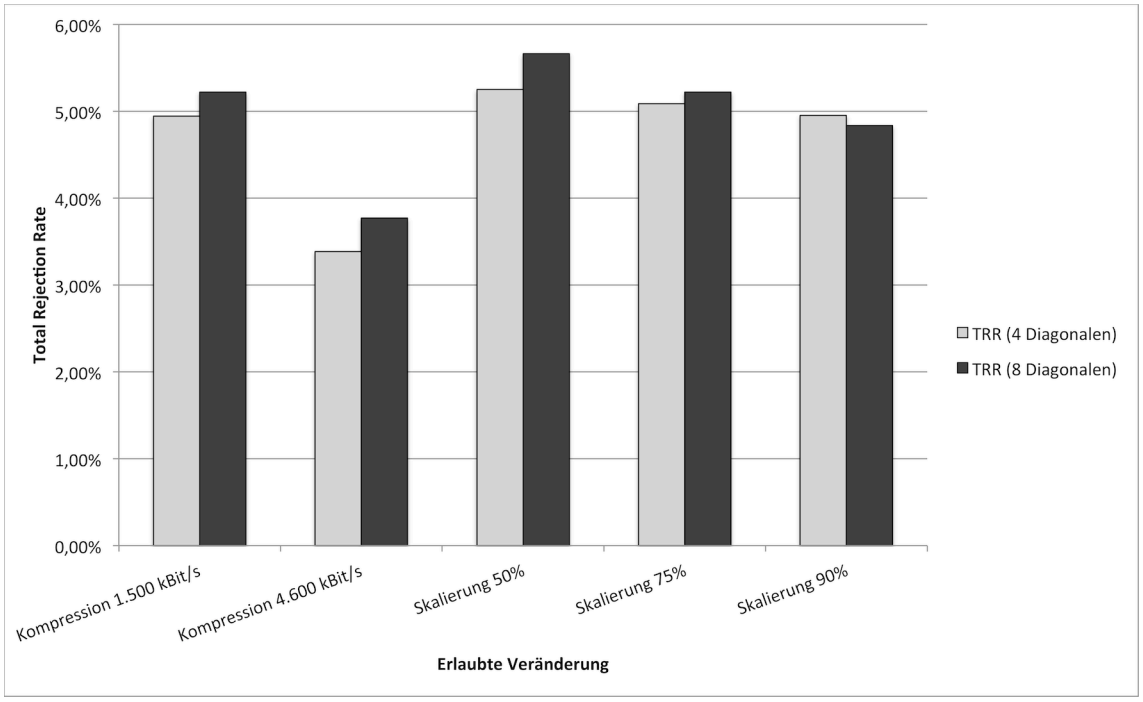


Abbildung 6.31: TRR des Moravec-Verfahrens ohne und mit Verwendung zusätzlicher Diagonalen

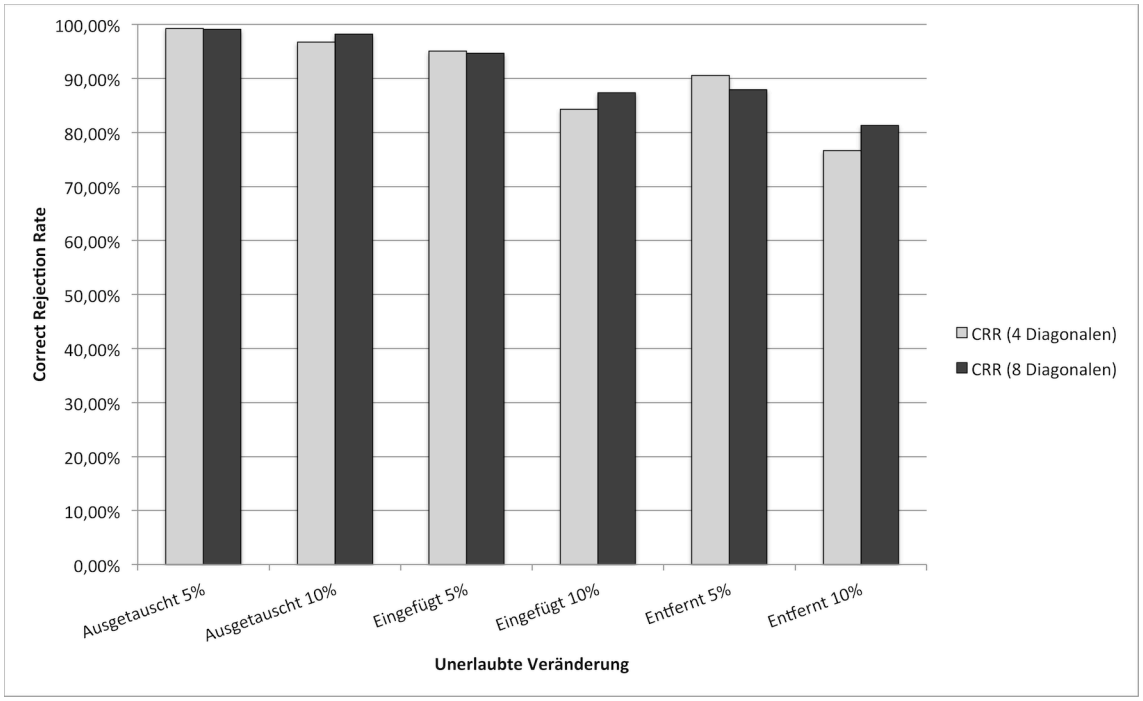


Abbildung 6.32: CRR des Moravec-Verfahrens ohne und mit Verwendung zusätzlicher Diagonalen

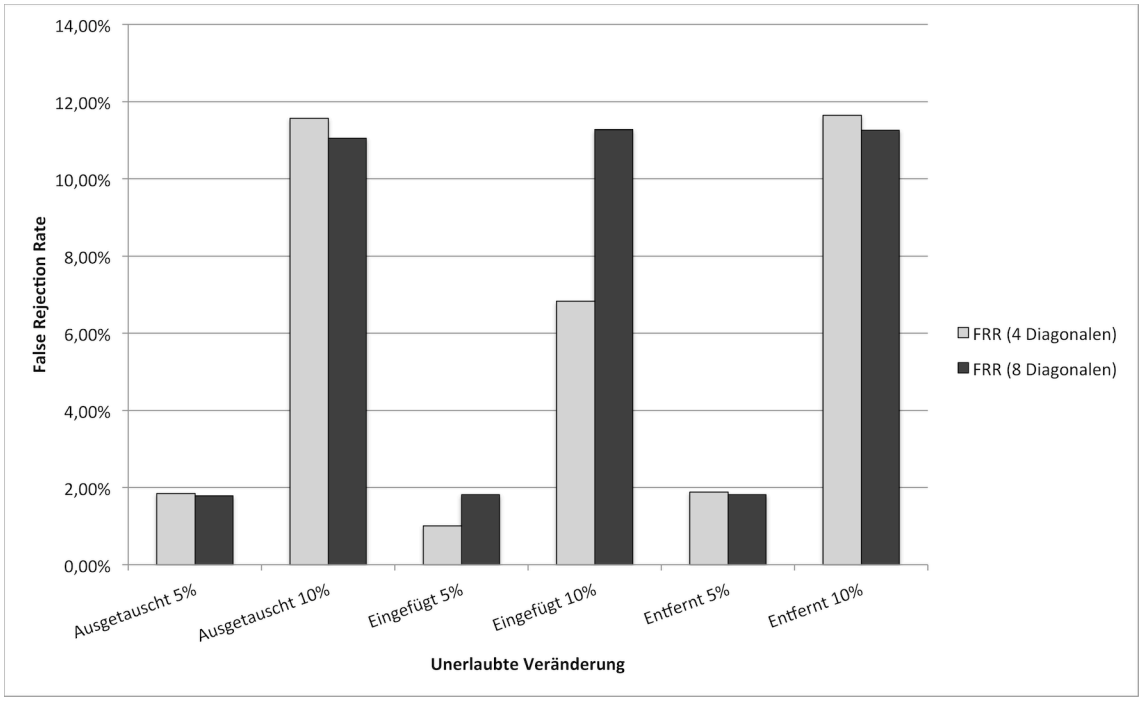


Abbildung 6.33: FRR des Moravec-Verfahrens ohne und mit Verwendung zusätzlicher Diagonalen

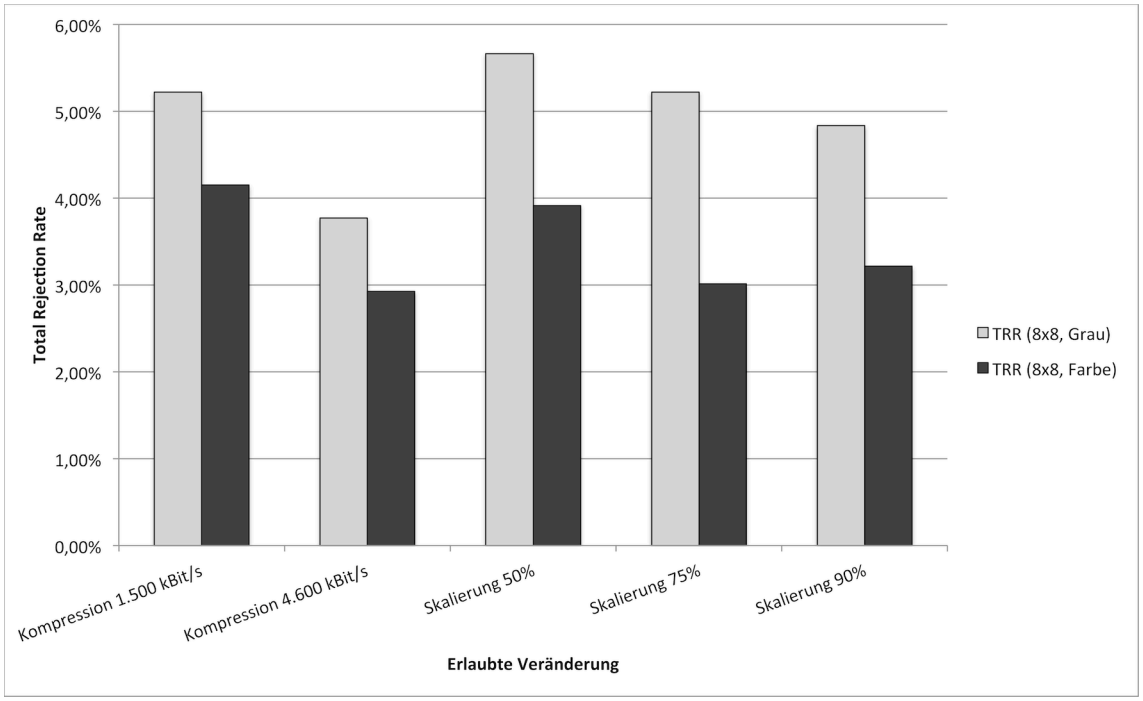


Abbildung 6.34: TRR des Moravec-Verfahrens unter Verwendung von Grau- und Farbwerten

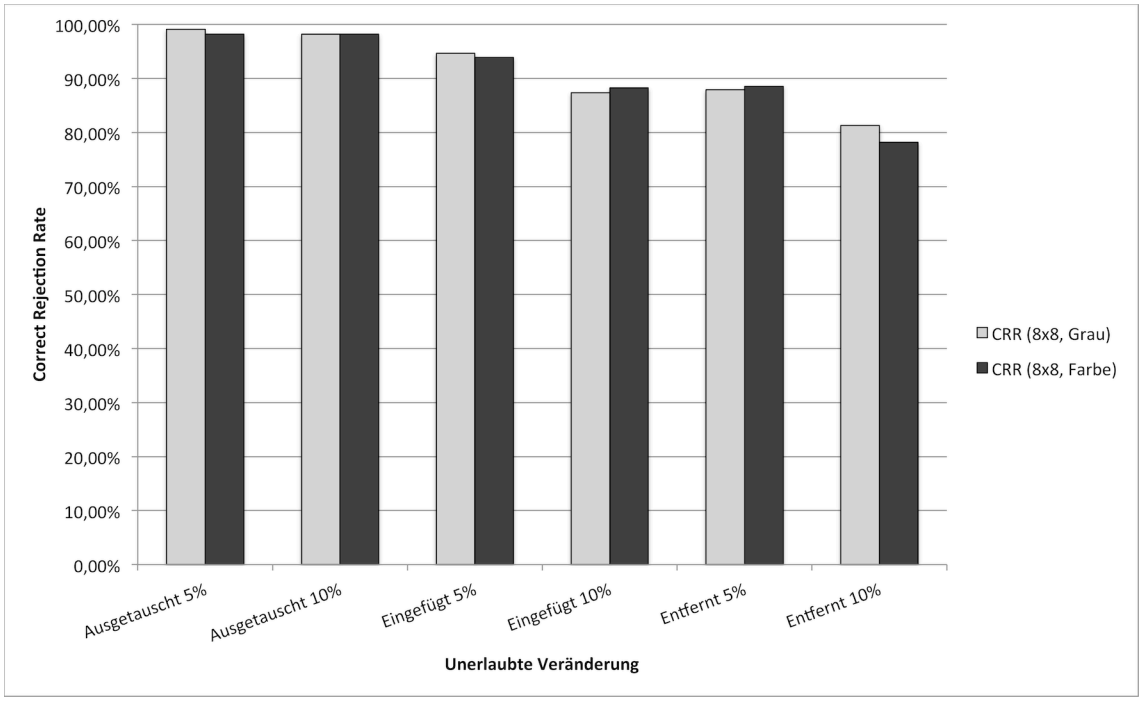


Abbildung 6.35: CRR des Moravec-Verfahrens unter Verwendung von Grau- und Farbwerten

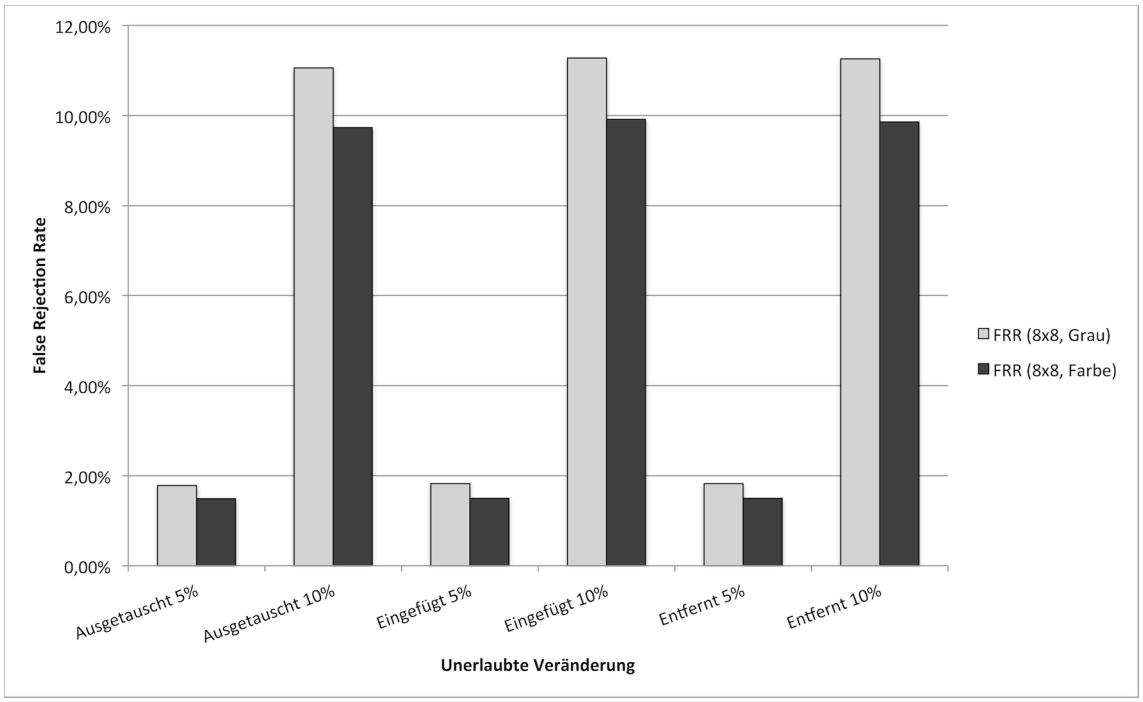


Abbildung 6.36: FRR des Moravec-Verfahrens unter Verwendung von Grau- und Farbwerten

6.4 Scale Invariant Feature Transform

Eine Alternative zum Interest-Operator von Moravec ist die Verwendung der Scale Invariant Feature Transform (SIFT), welche von Lowe in [Low04] vorgestellt wird. Sie berechnet ebenfalls aus Grauwertbildern markante Punkte, die SIFT-Features. Ein Punkt wird dann als markant eingestuft, wenn er sich von seinem Hintergrund abhebt. Die detektierten SIFT-Features können zur Objekterkennung verwendet werden, da sie selbst nach Drehung, Skalierung und Verschiebung wiedergefunden werden. Eine weitere Möglichkeit ist die Verwendung von SIFT-Features in der Bildverarbeitung. Zur Erstellung von Bildern mit hoher Auflösung werden Einzelbilder mit geringerer Auflösung zusammengefügt. Durch die Erkennung von markanten Punkten unterstützt SIFT das so genannte Super-Resoution-Verfahren [YY08]. Nach [Low04] lässt sich das Verfahren zum Erkennen von Features in vier Schritte unterteilen.

1. Bestimmung der Extrema in verschiedenen Skalierungsebenen

Im ersten Schritt wird das Eingangsbild in verschiedene Skalierungsebenen σ transformiert. Für ein Bild der Skalierungsebene $\sigma+1$ wird das Bild der Skalierungsebene σ auf die Hälfte seiner Auflösung reduziert. Für jede Skalierungsebene σ wird auf das Bild mehrfach ein Gauss-Weichzeichen-Filter angewendet. Nach jeder Anwendung wird das Differenzbild zum vorherigen Bild berechnet, welches über die Eigenschaft verfügt markante Stellen und Kanten hervorzuheben. Zur Bestimmung der Extrema werden die Minima und Maxima der einzelnen Skalierungsebenen σ berechnet. Ein Punkt stellt ein Extremum dar, wenn er innerhalb einer 3×3 - Nachbarschaft und zusätzlich innerhalb seiner benachbarten Differenzbilder ein Maximum oder Minimum repräsentiert.

2. Lokalisierung der Features

Im zweiten Schritt werden potentielle Features eliminiert, die gegen Bildrauschen nicht robust sind. Sie liegen oftmals an Kanten oder haben nur einen geringen Kontrast. Darüber hinaus werden den resultierenden Features genaue Koordinaten und eine Skalierungsebene zugewiesen.

3. Berechnung der Ausrichtung

Um eine Drehungsinvarianz zu erreichen, wird im dritten Schritt jedem detektierten Feature eine Ausrichtung zugewiesen, die im vierten Schritt verwendet wird. Dazu wird aus den umliegenden Bildpunkten eines Features ein Histogramm der Ausrichtungen berechnet. Jede Ausrichtung wird einer von 36 Zonen zugeordnet, die jeweils 10° umfassen. Aus der Zone mit den meisten Ausrichtungen werden die drei stärksten Gradienten ausgewählt, aus denen mittels Interpolation die Ausrichtung des Features berechnet wird.

4. Berechnung des Feature-Vektors

Im letzten Schritt wird jedem Feature ein Merkmalsvektor mit 128 Einträgen zugewiesen. Zur Berechnung des Vektors wird die Umgebung des Features in 4×4 quadratische Zonen unterteilt. In jeder dieser Zonen werden die berechneten Ausrichtungen der analysierten Bildpunkte jeweils einer von acht Hauptrichtungen zugeordnet. Um gegenüber Belichtungsveränderungen robust zu sein, werden die Vektoreinträge auf die Länge des größten Eintrags normalisiert.

Abbildung 6.37 stellt beispielhaft die Funktionalität der Scale Invariant Feature Transform dar. Das Bild wurde mit Hilfe der Demonstrationssoftware von David Lowe (University of British Columbia) bearbeitet[1]. Auf das Bild wurde eine SIFT durchgeführt und die SIFT-Features mit ihren Richtungsvektoren angezeigt. Die Mehrzahl der SIFT-Features befinden sich in Abbildung 6.37 im unteren und mittleren Bereich.

Abbildung 6.37: Darstellung der Richtungsvektoren für SIFT-Features

6.4.1 Generierung des Merkmalsvektors und Einbettung

Sei $F = \{f_1, \ldots, f_N\}$ die Menge aller Videoframes in einem Video. Für jedes Frame f_n wird sein inhalts-beschreibender Merkmalsvektor V_n in den folgenden Schritten berechnet. Analog zu den vorherigen beiden Verfahren wird zunächst das Frame f_n auf eine feste Größe skaliert (ebenfalls 128×128 Pixel) um eine feste Länge des Merkmalsvektors zu erreichen. Danach wird das Bild mittels eines Weichzeichen-Filters geglättet und die SIFT-Punkte berechnet. Eine Tiefpass-Filterung wird nicht durchgeführt, da sie die Genauigkeit der Punkt-Erkennung negativ beeinflussen kann.

Wir unterteilen die Blöcke in Gruppen $G_n = \{G_{n,1}, \ldots, G_{n,O}\}$. Ein Vektorbit des Merkmalsvektors $V_n = v_{n,1} \| \ldots \| v_{n,M}$ des Frames f_n wird wie folgt gebildet:

- Bildet die Summe der SIFT-Punkte in $b_{n,m}$ innerhalb seiner Gruppe $G_{n,o}$ ein Maximum, dann wird sein zugehöriges Vektorbit $v_{n,m}$ auf 1 gesetzt.

[1] http://www.cs.ubc.ca/~lowe/keypoints/ (Aufruf am 10.03.2013)

- Ist die Summe der SIFT-Punkte von $b_{n,m}$ innerhalb seiner Gruppe $G_{n,o}$ größer als die Hälfte aller Blöcke in $G_{n,o}$, dann wird sein zugehöriges Vektorbit $v_{n,m}$ auf 1 gesetzt.

- Wird keine der vorherigen Bedingungen erfüllt, so wird sein zugehöriges Vektorbit $v_{n,m}$ auf 0 gesetzt.

Der Merkmalsvektor V_n wird mit einem robusten Wasserzeichenverfahren in das Frame f_{n+1} eingebettet. Zusätzlich wird ein eindeutiger, kontinuierlich steigender Frameindex dem Merkmalsvektor angehängt, um Manipulationen an der Zeitachse zu erkennen. In [TS10] haben wir ein ähnliches Verfahren vorgestellt. Allerdings werden ähnlich wie in [DTS04] Relationen zwischen Blockpaaren verschiedener Frames hergestellt, was zu einer Verschlechterung der Sensitivität führte. Daher verwenden wir hier die Variante, dass innerhalb einer Blockgruppe die Maxima bestimmt werden.

Abbildung 6.38 stellt das Verfahren zur Generierung des Merkmalsvektors mit Hilfe der SIFT noch einmal schematisch dar. Das Frame f_n wird vorverarbeitet und seine Interest-Werte werden berechnet. Anschließend werden Gruppen gebildet und auf deren Basis der Merkmalsvektor berechnet. Dieser Vektor V_n wird anschließend unter der Kontrolle eines geheimen Schlüssels K robust in das Frame f_{n+1} eingebettet.

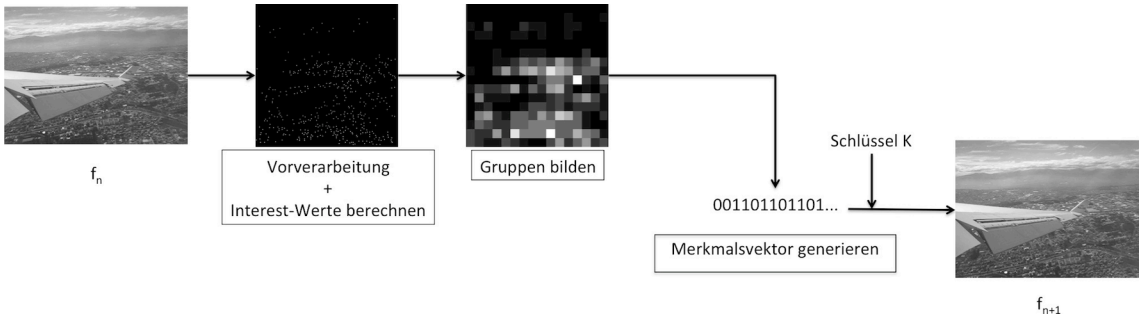


Abbildung 6.38: Merkmalsgenerierung für das Verfahren mit der SIFT

In Abbildung 6.39 stellen wir abschließend ein Frame seinen erkannten SIFT-Interestwerten gegenüber. Wie darauf zu erkennen ist, befindet sich die Mehrzahl der SIFT-Interestwerte im unteren und mittleren Bereich des Bildes. Der Himmel ist für das SIFT-Verfahren (wie auch für das Moravec-Verfahren) nicht signifikant.

6.4.2 Auslesen des Merkmalsvektors und Verifikation

Im Ausleseprozess wird wiederum der Merkmalsvektor V_n aus dem Frame f_{n+1} ausgelesen. Wir berechnen, wie in Abschnitt 6.4.1 beschrieben, den aktuellen und möglicherweise modifizierten Merkmalsvektor $\tilde{V}_n$ aus dem Frame f_n. Mittels der in Abschnitt 6.2.2 vorgestellten Entscheidungshilfen kann analysiert werden, an welchen Positionen das Video manipuliert wurde. Um inhalts-erhaltende Maßnahmen

Abbildung 6.39: Gegenüberstellung eines Frames mit seinen SIFT-Interestwerten

von inhalts-verändernden Maßnahmen zu unterscheiden, verwenden wir den in Abschnitt 6.2.2 eingeführten zeitlichen Filter.

6.4.3 Analyse des Merkmals

In diesem Abschnitt analysieren wir die Robustheit und Sensitivität des Merkmals. Wir verwenden zur Analyse das bekannte Testvideo und analysieren verschiedene Parameter hinsichtlich der bekannten inhalts-erhaltenden und inhalts-verändernden Maßnahmen. Die besten Parametersätze mit verschiedenen Block- und Gruppengrößen, die in den Abbildungen 6.40, 6.41 und 6.42 dargestellt sind, können Tabelle 6.8 entnommen werden.

Blockgröße	Gruppengröße	Filterlänge
8×8	2	1s
8×8	4	3s
8×8	8	4s
16×16	2	2s
16×16	4	3s
16×16	8	4s

Tabelle 6.8: Parametersätze für die verschiedenen Block- und Gruppengrößen

Aus den Testergebnissen können wir folgende Erkenntnisse ziehen:

- Die Blockgröße hat einen signifikanten Einfluss auf die Robustheit des Verfah-

rens. Bei einer Blockgröße von 8×8 Pixeln betrug die durchschnittliche Total Rejection Rate (TRR) 4,93%. Mit einer Blockgröße von 16×16 Pixeln sank sie auf 1,15%.

- Umgekehrt verhält es sich mit der Sensitivität des Verfahrens. Bei einer geringeren Blockgröße konnte eine höhere durchschnittliche Correct Rejection Rate (CRR) erzielt werden (76,18%) als bei der größeren Blockgröße (47,32%).
- Der Einfluss der Gruppengröße ist wieder uneinheitlich. Mit einer mittleren Gruppengröße erzielten wir bei beiden Blockgrößen die besten Ergebnisse hinsichtlich der Sensitivität (79,71% bzw. 58,30%).

Fazit: Das SIFT-Verfahren erwies sich als nicht so robust und sensitiv, wie das Entropie- und das Moravec-Verfahren.

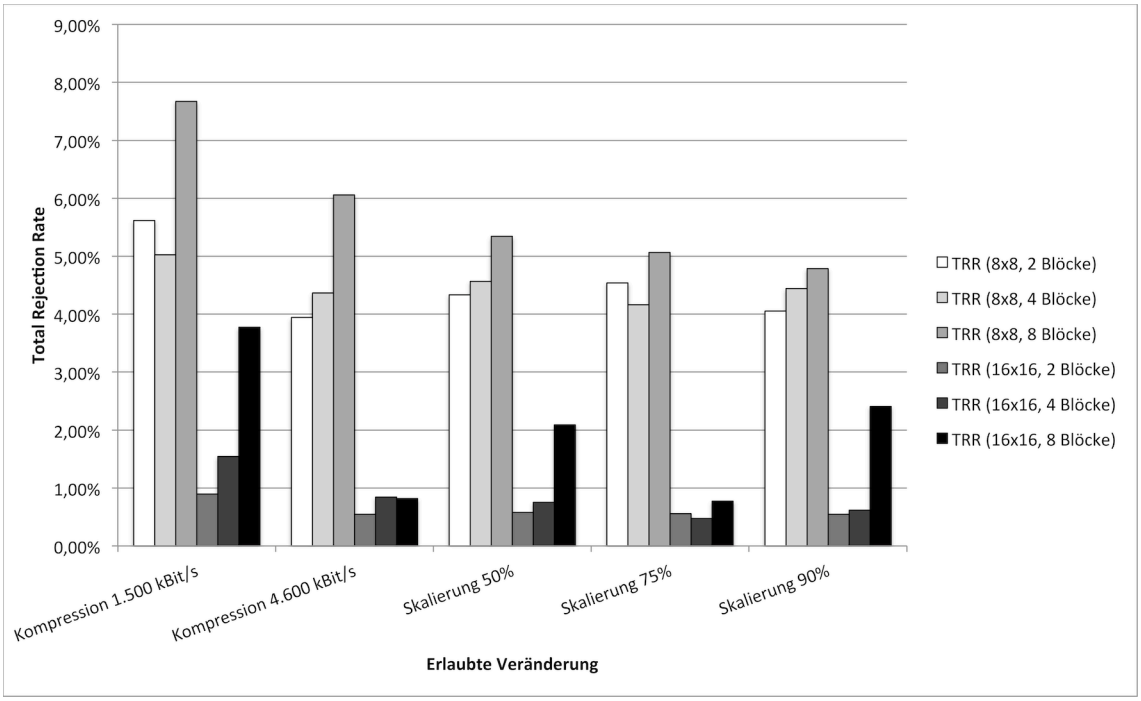


Abbildung 6.40: TRR des SIFT-Verfahrens gruppiert nach Block- und Gruppengröße

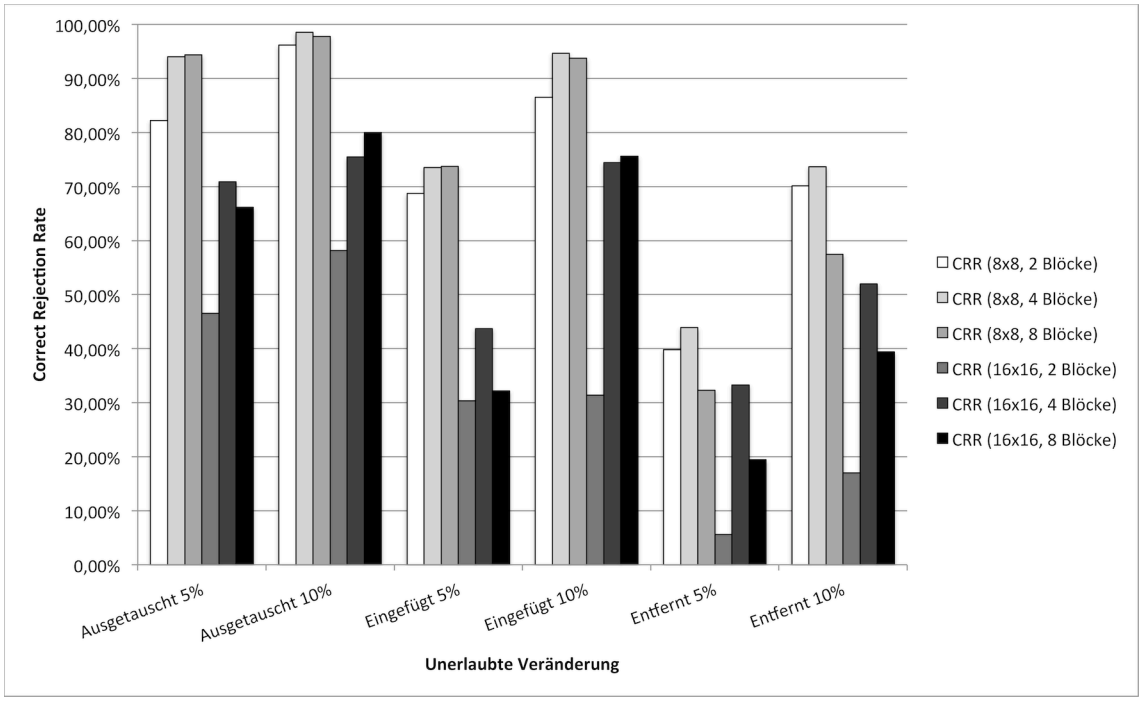


Abbildung 6.41: CRR des SIFT-Verfahrens gruppiert nach Block- und Gruppengröße

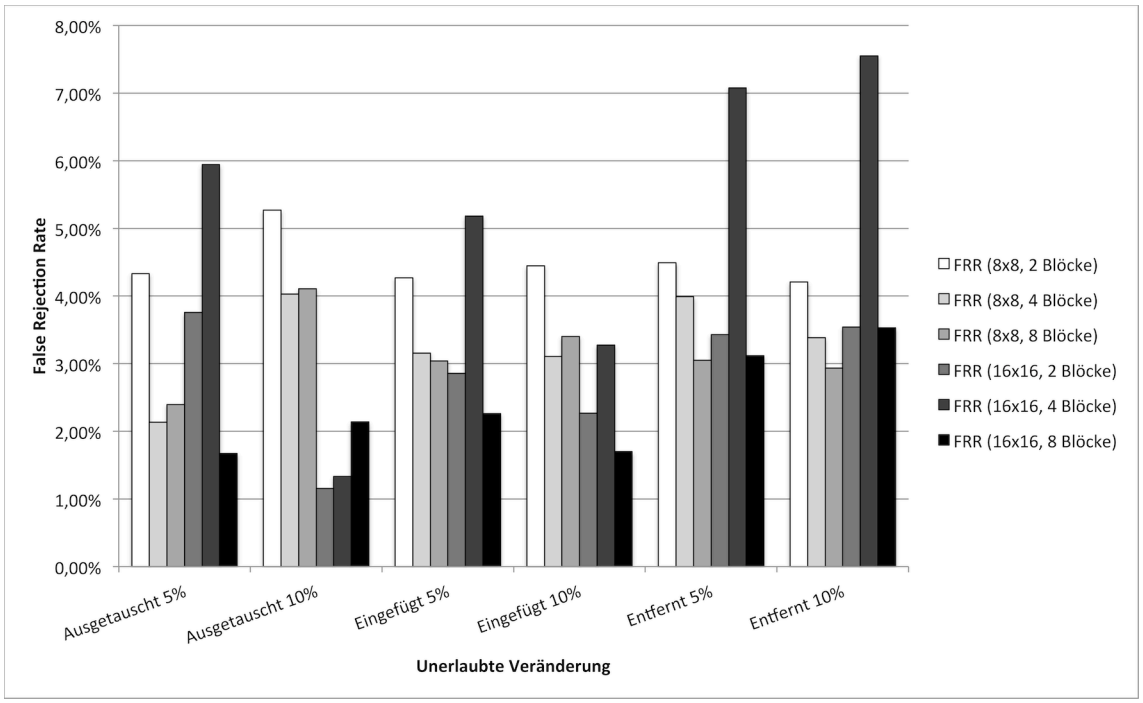


Abbildung 6.42: FRR des SIFT-Verfahrens gruppiert nach Block- und Gruppengröße

6.5 Zusammenfassung

Wir haben in diesem Kapitel vier verschiedene Verfahren vorgestellt, die inhaltliche und nicht-inhaltliche Abhängigkeiten auf binäre Merkmalsvektoren abbilden. Gleichzeitig wurde jedem Merkmalsvektor ein eindeutiger Index zugewiesen, um Manipulationen an der Zeitachse zu erkennen. Die Verfahren wurden unter einheitlichen Bedingungen hinsichtlich ihrer Robustheit gegenüber inhalts-erhaltenden und hinsichtlich ihrer Sensitivität gegenüber inhalts-verändernden Maßnahmen untersucht. Damit wurde eine Vergleichbarkeit der verschiedenen Verfahren erzielt.

In Kapitel 4.3 stellten wir verschiedene Anforderungen an das inhalts-beschreibende Merkmal. Bei den Manipulationen an Einzelbildern wollten wir Manipulationen an Objekten, Veränderungen der Helligkeit und der Farbe erkennen können. Die Manipulationen an Objekten wurden aufgrund der besseren Automatisierbarkeit durch Blockmanipulationen simuliert. Die manipulierten Blöcke mit einer Größe von 10% der Gesamtfläche passten in das Gruppenraster des Entropie-Verfahrens, des Moravec-Verfahrens und des SIFT-Verfahrens. Demgegenüber befanden sich die manipulierten Blöcke mit einer Größe von 5% der Gesamtfläche innerhalb der Blockgruppen. Die dadurch entstandenen schärferen Kanten wirkten sich jedoch nicht verbessernd auf die Erkennbarkeit der Manipulationen aus. Daher gehen wir davon aus, dass auch die Manipulation von polygonen Flächen, welche der Manipulation von realen Objekten besser entspricht, ähnliche Ergebnisse erzielen würde. Die Veränderungen wurden vom Entropie-Verfahren und vom Moravec-Verfahren gut erkannt. Ausstehend sind Untersuchungen über die Sensitivität hinsichtlich der Manipulationen an der Helligkeit und Farbveränderungen. Dies ist die Aufgabe weiterer Forschung. Hinsichtlich der Erkennung von Manipulationen an der Zeitachse wurde in allen Verfahren ein kontinuierlich steigender Frameindex eingeführt, der dem Merkmalsvektor angehängt wurde. Die Erkennbarkeit der Manipulationen hängt hierbei nicht vom inhalts-beschreibenden Merkmal sondern von der Robustheit des zugrundeliegenden Wasserzeichenverfahrens ab.

Die Verfahren wurden auch hinsichtlich ihrer Robustheit getestet. Sie wurden verlustbehafteter Kompression, Skalierung und Formatumwandlung unterzogen. Hier zeigte sich eine besonders gute Robustheit des Energiedifferenz-Verfahrens. Auch das Entropie- und das Moravec-Verfahren erwiesen sich als robust. Weitere Untersuchungen hinsichtlich der anderen in Kapitel 4.2 genannten Forderungen sind ebenfalls die Aufgabe weiterer Forschung.

Das Entropie-Verfahren zeigte sich insgesamt als am geeignetsten. Das Moravec-Verfahren zeigte sich ebenso geeignet hinsichtlich seiner Sensitivtät, war jedoch nicht vergleichbar robust. Daher werden wir in der folgenden abschließenden Evaluierung das Entropie-Verfahren als inhalts-beschreibendes Merkmal verwenden.

Kapitel 7

Evaluierung

In diesem Kapitel evaluieren wir den in Kapitel 4 vorgestellten Konzeptentwurf - die Kombination eines Verfahrens zur Generierung von Merkmalsvektoren und eines robusten Wasserzeichenverfahrens.

7.1 Erweiterte Analyse des Entropie-Verfahrens

Für die abschließende Evaluierung haben wir 15 Videos verwendet. Aus Gründen der Vergleichbarkeit verfügten die Videos über gleiche Eigenschaften:

- Auflösung: 720×576 Pixel
- Bitrate: 9.200 kBit/s
- Framerate: 25 Frames/s
- Spieldauer: 90 Sekunden

Die Videos entstammen 5 verschiedenen Quellen mit unterschiedlichen Charakteristiken:

- Trailer: Trailer zum Film „American Pie“ (Quelle: Constantin Film). Schnelle Schnitte, Szenen mit unterschiedlicher Belichtung
- Kindersendung 1 - 3: Kindersendung „Spur und Partner“ (Quelle: ORB). Teilweise schnelle Bewegungen, kräftige Farben, wenig Texturierung
- Landschaften 1 - 5: Film mit Landschaftsaufnahmen (Quelle: Firma MedienMotor). Langsame Kamerafahrten, wenig Bewegung, Letterbox

- Lehrvideo 1 - 4: Internes Video zum Thema Suchtbekämpfung (Quelle: Fraunhofer Gesellschaft). Ausschnitte aus einem Interview, einer Rede und Filmeinspielungen, wenig Bewegung, unterschiedliche Helligkeit und Farbgebung, Videos 2 - 4 enthielten eine Letterbox

- Soccer 1 - 2: Fußballspiel (Quelle: SPORTiV 1, Italien), viele Kameraschnitte, viel Bewegung, teils große plane Flächen

Während der Analyse in Kapitel 6 erzielte das Entropie-Verfahren mit dem Testvideo des Instituts für Rundfunktechnik (IRT) die besten Ergebnisse. Daher verwendeten wir dieses Verfahren um es mit den 15 Videos zu evaluieren. Bei den Videos, die eine Letterbox enthielten, wurde darauf geachtet, dass die Manipulationen im Sichtbereich durchgeführt wurden und nicht in der Letterbox selbst. Dies würde sonst die Ergebnisse verfälschen. Wieder wurden zunächst zwei verschiedene Blockgrößen (8×8 und 16×16 Pixel) und drei verschiedene Blockgruppengrößen (2, 4 und 8 Blöcke pro Gruppe) getestet. Die jeweils besten Parameter mit ausgewogener Robustheit und Sensitivität sind Tabelle 7.1 zu entnehmen.

Blockgröße	Gruppengröße	Quantisierungsfaktor QF	Filterlänge
8×8	2	25	3s
8×8	4	25	1s
8×8	8	50	3s
16×16	2	10	1s
16×16	4	25	3s
16×16	8	50	3s

Tabelle 7.1: Parametersätze für die verschiedenen Block- und Gruppengrößen

Die Ergebnisse für die verschiedenen Block- und Gruppengrößen sind in den Abbildungen 7.1, 7.2 und 7.3 dargestellt. Folgende Erkenntnisse haben wir aus den Ergebnissen gewonnen:

- Mit größeren Blöcken sinkt die Sensitivität des Merkmals. Mit einer Blockgröße von 8×8 Pixeln erreichen wir eine maximale durchschnittliche Correct Rejection Rate (CRR) von 82,52% bei einer Gruppengröße von 8 Blöcken. Demgegenüber sinkt die CRR auf 58,95% bei einer Blockgröße von 16×16 Pixeln und ebenfalls einer Gruppengröße von 8 Blöcken. Die Robustheit steigt minimal bei größeren Blöcken.

- Mit größeren Gruppen steigt die Sensitivität des Merkmals bei relativ konstanter Robustheit. Die CRR variiert bei 8×8 Pixel großen Blöcken zwischen 75,15% und 82,52% mit einer False Rejection Rate (FRR) zwischen 1,21% und 2,57%. Die Total Rejection Rate (TRR) liegt bei den gleichen Parametersätzen zwischen 0,51% und 1,54%.

- Im Vergleich zum Video des IRT sank die CRR bei 8×8 Pixel großen Blöcken um durchschnittlich 7,94% während gleichzeitig die FRR um 1,15% anstieg (16×16 Pixel: CRR um 5,67% gestiegen, FRR um 0,44% gesunken). Die TRR stieg bei 8×8 Pixel großen Blöcken um durchschnittlich 0,53% während sie bei 16×16 Pixel großen Blöcken um 0,54% sank.

Da eine Merkmalsvektorlänge von 256 Bit nicht robust eingebettet werden kann, entschieden wir uns trotz geringerer Sensitivität für den Einsatz der größeren Blöcke mit einer Auflösung von 16×16 Pixeln. Damit sinkt die Länge des Vektors auf 64 Bit pro Frame.

In einer weiteren Testreihe untersuchten wir den Einfluss des Quantisierungsfaktors QF. Dafür verwendeten wir die Parametersätze aus Tabelle 7.2. Die Ergebnisse sind den Abbildungen 7.4 (TRR), 7.5 (CRR) und 7.6 (FRR) dargestellt. Folgende Erkenntnisse sind daraus abzuleiten:

- Der Quantisierungsfaktor QF reagiert uneinheitlich hinsichtlich der Robustheit. Während ein niedriger Quantisierungsfaktor $QF = 1$ eine durchschnittliche TRR von 1,44% hat, liegt sie bei einem leicht höheren Quantisierungsfaktor $QF = 10$ bei durchschnittlich nur noch 0,46%.
- Mit steigendem Wert von QF erhöht sich auch die Sensitivität. Die CRR, das Maß für die Sensitivität bewegt sich durchschnittlich zwischen 53,13% und 58,95%.
- Die besten Resultate wurden mit einem Quantisierungsfaktor $QF = 50$ erzielt.

In einer weiteren Testreihe untersuchten wir die Filterlänge. Dafür wurden die Parametersätze aus Tabelle 7.3 verwendet. Die Ergebnisse sind in den Abbildungen 7.7 (TRR), 7.8 (CRR) und 7.9 (FRR) dargestellt. Folgendes haben wir festgestellt:

- Die Filterlänge wirkt sich nicht einheitlich auf die Robustheit und Sensitivität aus.
- Die CRR bewegt sich durchschnittlich zwischen 44,23% (4 Sekunden) und 58,95% (3 Sekunden).
- Die TRR liegt zwischen durschnittlich zwischen 0,53% (1 Sekunde) und 1,20% (5 Sekunden). Im Gegensatz zu den Ergebnissen in Kapitel 6.2.4 wirkt sich hier der zeitliche Filter eher negativ auf die Robustheit aus.
- Die besten Resultate erzielten wir mit einer Filterlänge von 3 Sekunden.

Für den besten Parametersatz haben wir in den Abbildungen 7.10 (Blockaustausch), 7.11 (Blöcke eingefügt) und 7.12 (Blöcke entfernt) die Testergebnisse noch einmal

nach den 15 Videos aufgeschlüsselt. Der beste Parametersatz hatte eine Gruppengröße von 8 Blöcken, einen Quantisierungsfaktor QF von 50 und eine Filterlänge von 3 Sekunden. Wie ersichtlich ist, zeigt das Verfahren eine gute Sensitivität bei Manipulationen von 10%. Schwächen zeigte das Verfahren, wenn die Manipulationen geringer waren, insbesondere beim Entfernen von Blöcken der Größe 5%.

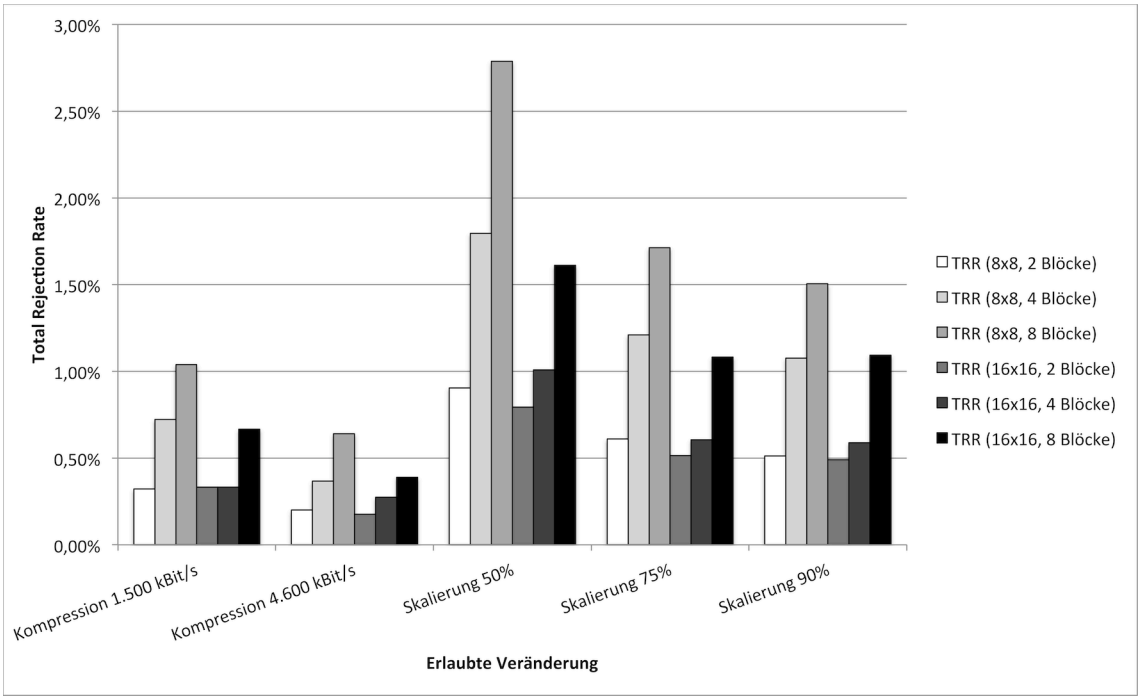


Abbildung 7.1: TRR des Entropie-Verfahrens gruppiert nach Block- und Gruppengröße

Quantisierungsfaktor QF	Gruppengröße	Filterlänge
1	2	1s
10	2	1s
25	4	3s
50	8	3s

Tabelle 7.2: Parametersätze für die Werte von QF

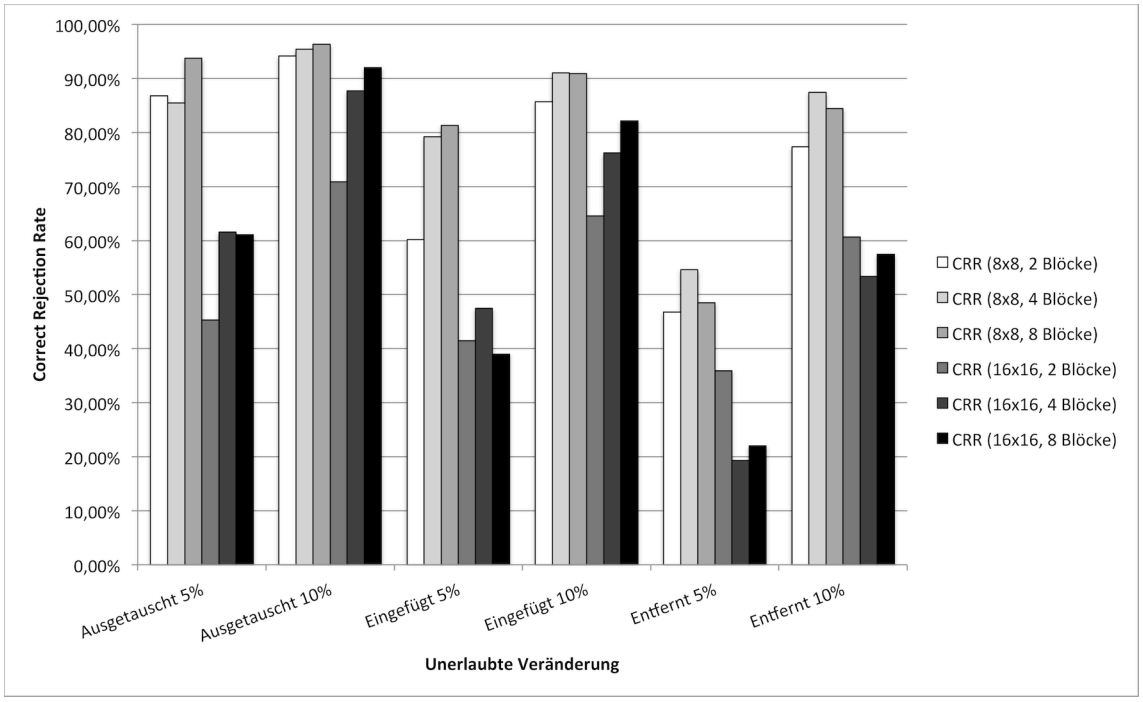


Abbildung 7.2: CRR des Entropie-Verfahrens gruppiert nach Block- und Gruppengröße

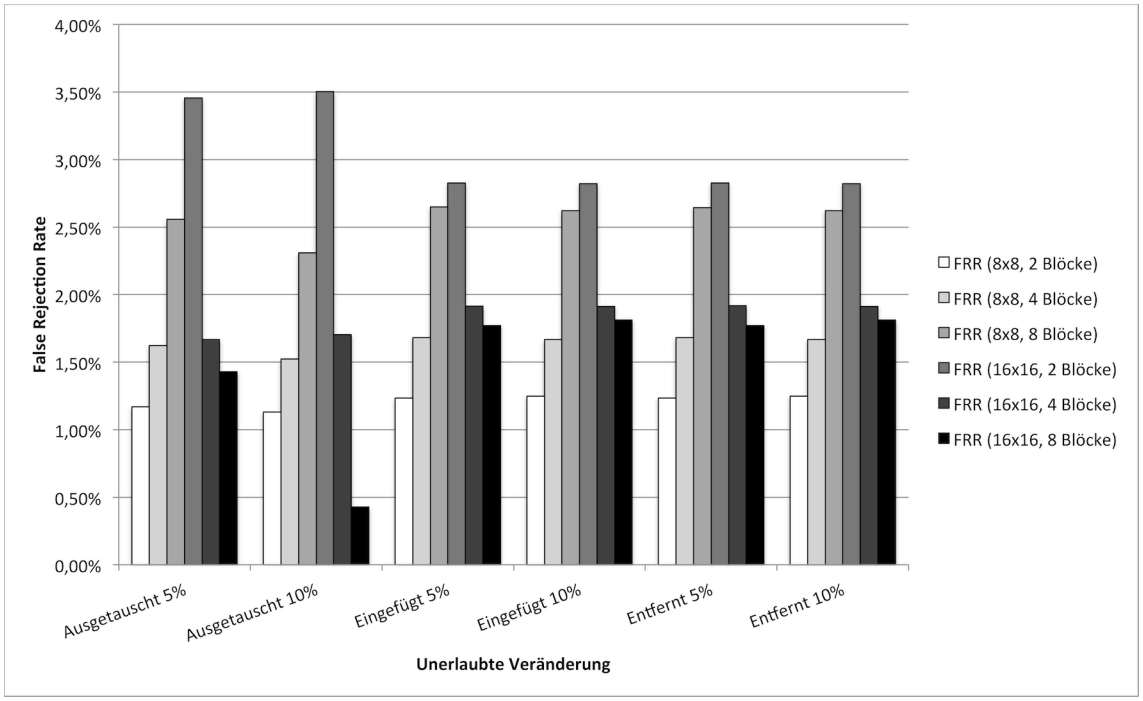


Abbildung 7.3: FRR des Entropie-Verfahrens gruppiert nach Block- und Gruppengröße

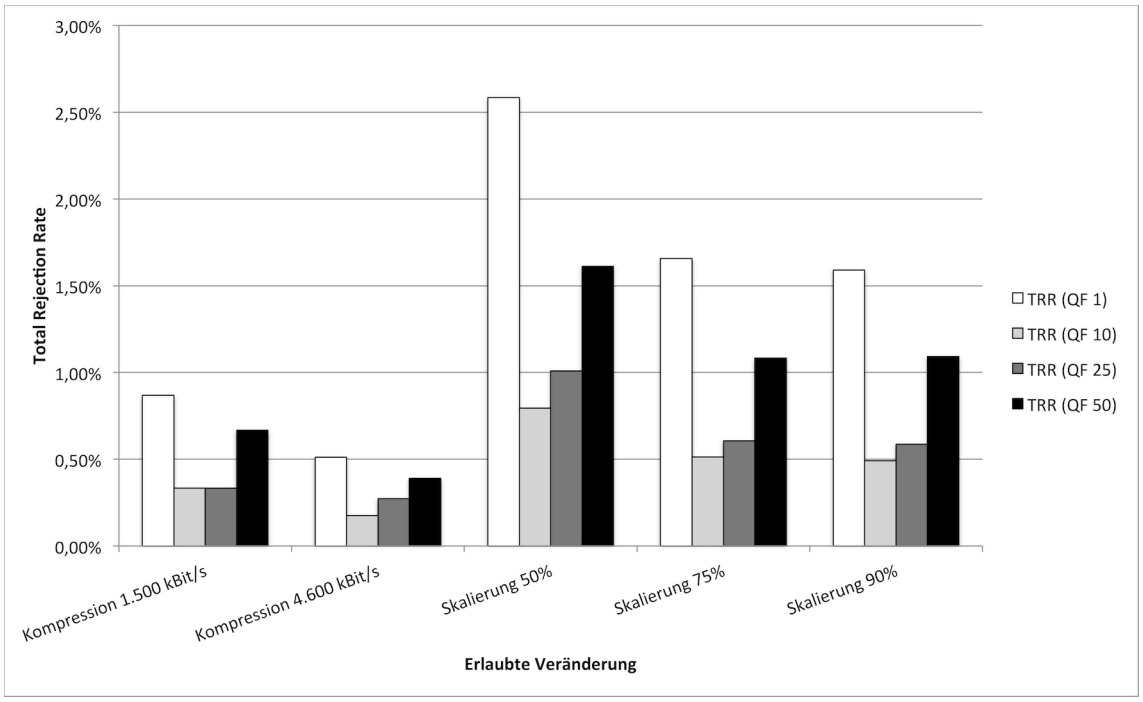


Abbildung 7.4: TRR des Entropie-Verfahrens gruppiert nach Quantisierungsfaktor QF

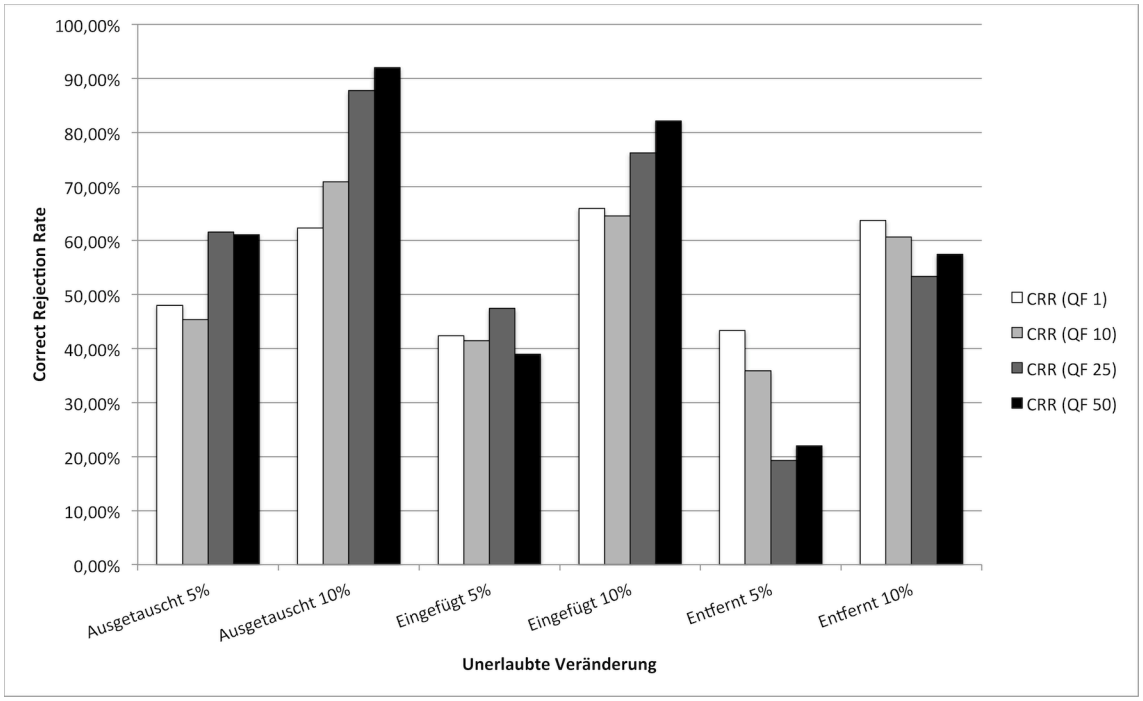


Abbildung 7.5: CRR des Entropie-Verfahrens gruppiert nach Quantisierungsfaktor QF

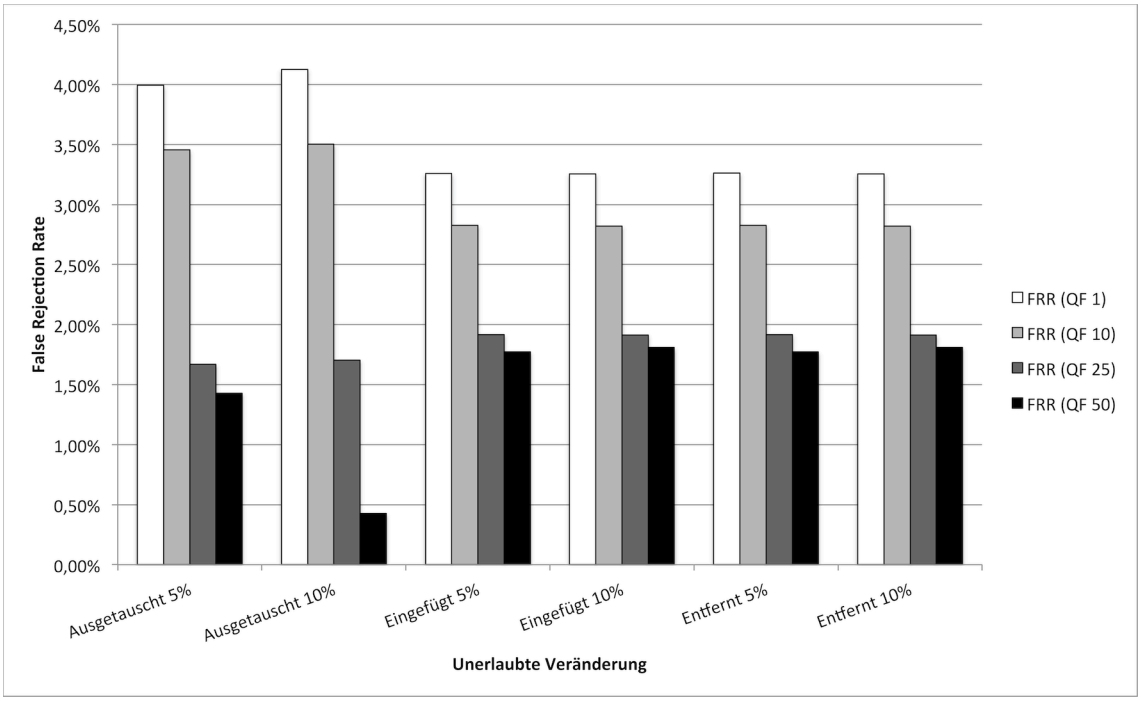


Abbildung 7.6: FRR des Entropie-Verfahrens gruppiert nach Quantisierungsfaktor QF

Filterlänge	Gruppengröße	Quantisierungsfaktor QF
1s	4	50
2s	4	50
3s	8	50
4s	2	25
5s	4	50

Tabelle 7.3: Parametersätze für die Filterlängen

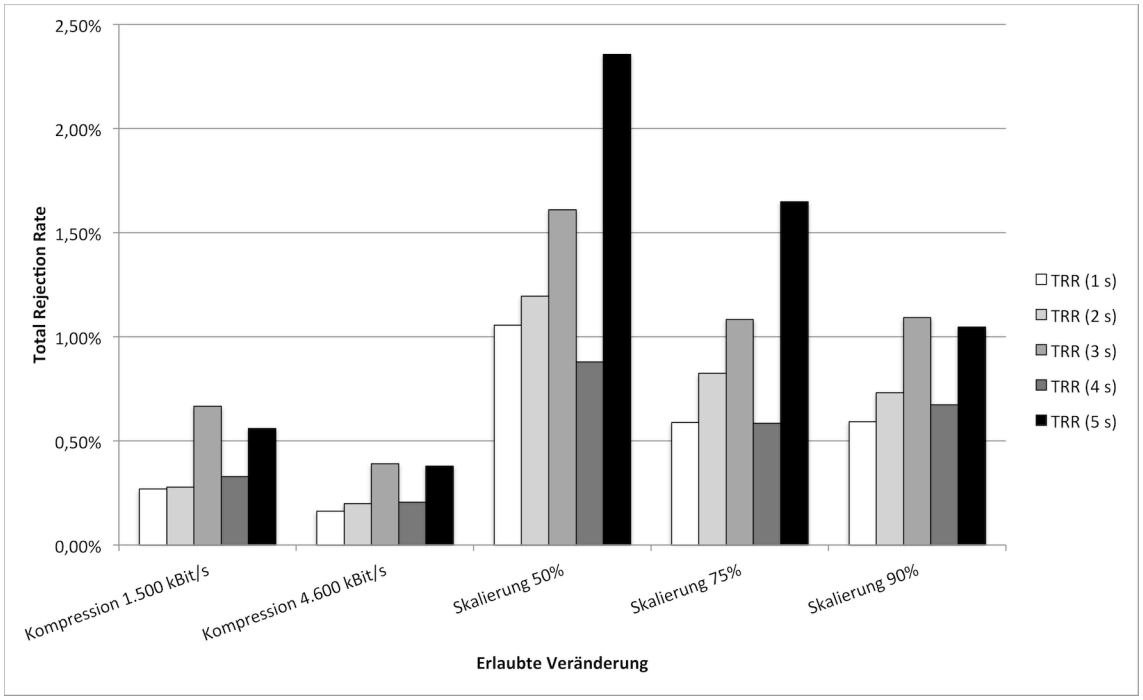


Abbildung 7.7: TRR des Entropie-Verfahrens gruppiert nach Filterlänge

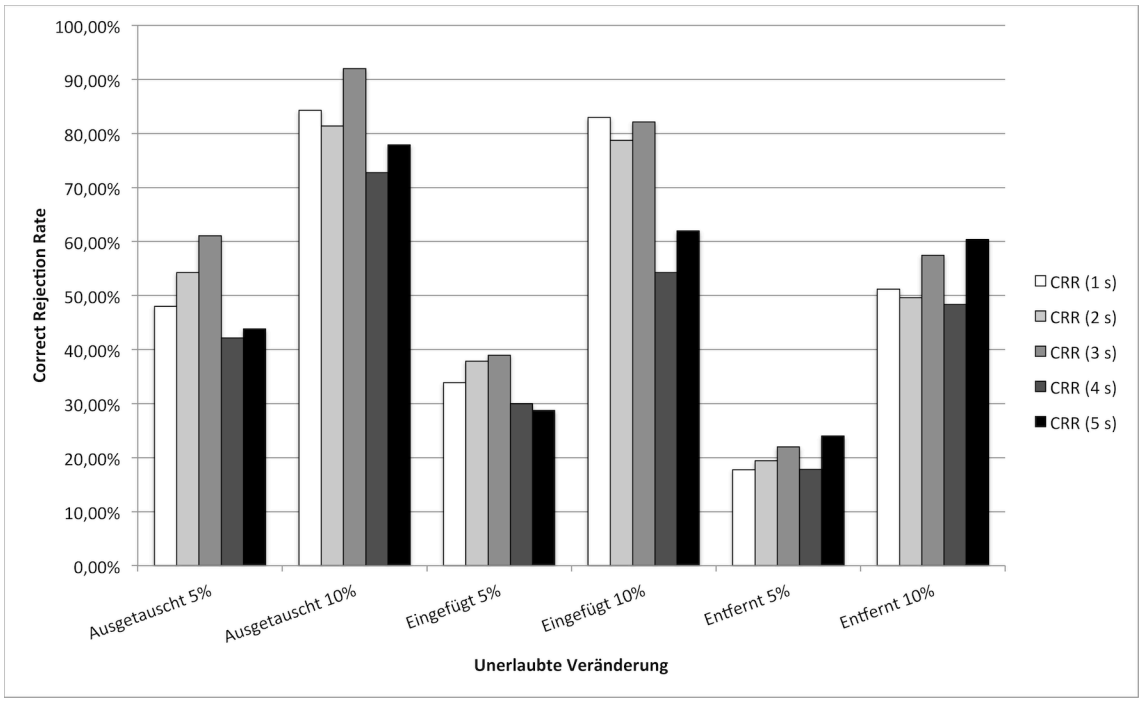


Abbildung 7.8: CRR des Entropie-Verfahrens gruppiert nach Filterlänge

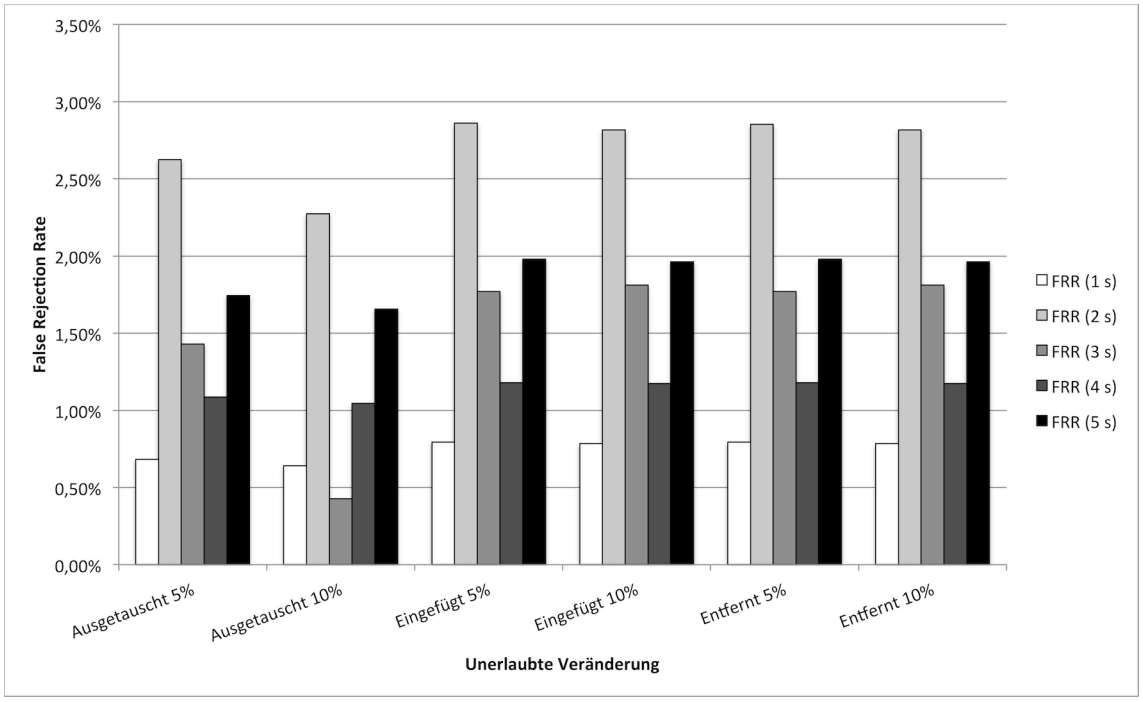


Abbildung 7.9: FRR des Entropie-Verfahrens gruppiert nach Filterlänge

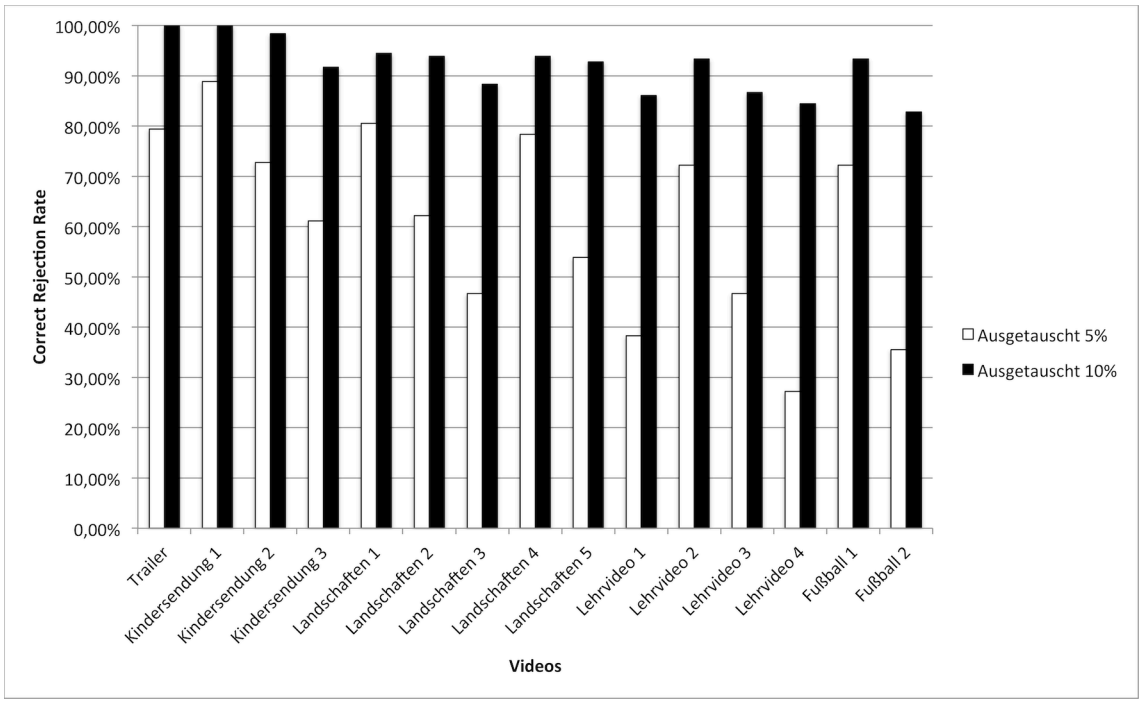


Abbildung 7.10: CRR nach dem Austausch von Blockpaaren gruppiert nach Video

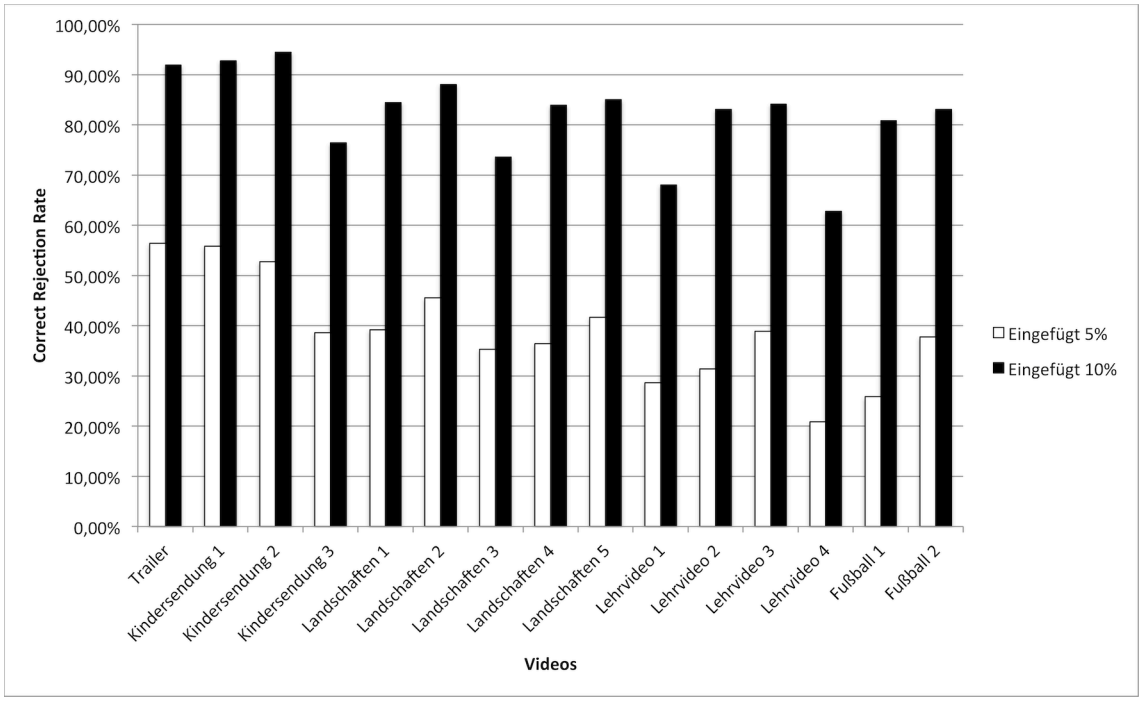


Abbildung 7.11: CRR nach dem Einfügen von Blöcken gruppiert nach Video

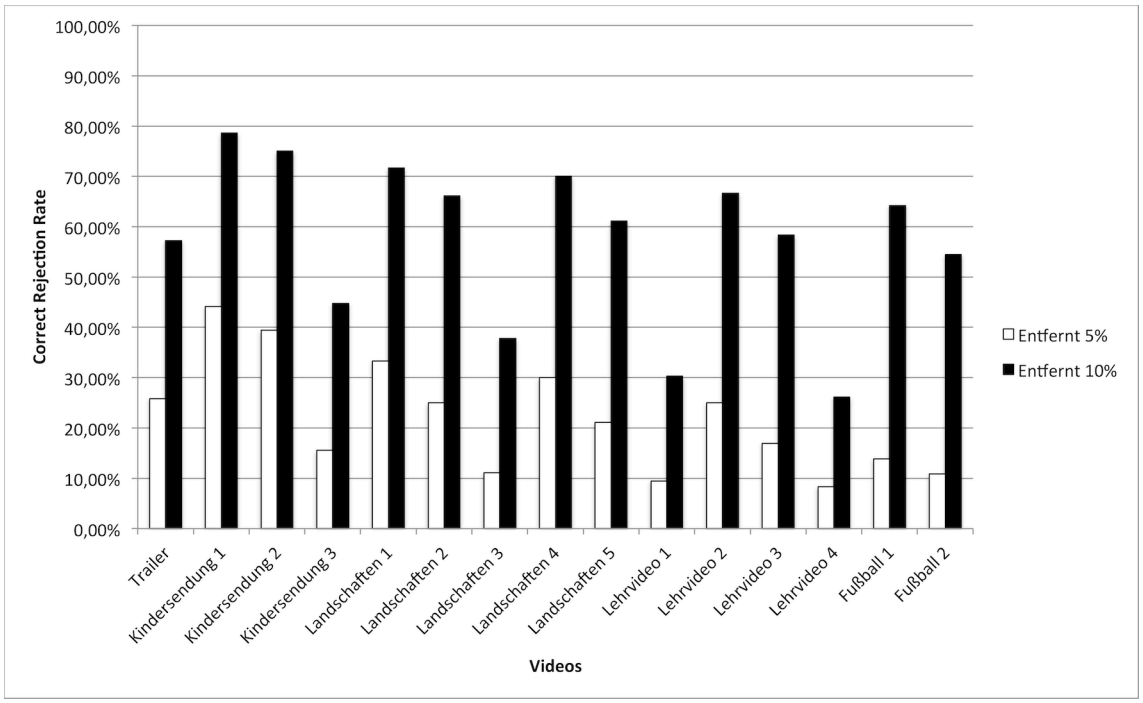


Abbildung 7.12: CRR nach dem Entfernen von Blöcken gruppiert nach Video

7.2 Erweiterte Analyse des robusten Wasserzeichens

Um die Auslesequalität des zugrundeliegenden robusten Wasserzeichenverfahrens zu analysieren, wurden die 15 Videos mit den Merkmalsvektoren des oben genannten Parametersatzes markiert. Als Einbettungsverfahren verwendeten wir das in Kapitel 5.1 vorgestellte Verfahren. Die Blockgröße wurde auf 8×8 Pixel festgelegt, da sie über die beste Robustheit verfügte. Um das Verfahren zur Generierung der Merkmalsvektoren nicht zu beeinflussen, wurden die ersten 6 DCT-Koeffizienten von der Markierung ausgeschlossen. Da mit einer Einbettungsrate von 64 Bit/s nicht der gesamte Merkmalsvektor aus Robustheitsgründen komplett in jedes Frame eingebettet werden konnte, wurde die Nachricht in 5 Teile gesplittet. Dazu wurde das in Kapitel 5.1.4 vorgestellte Verfahren verwendet. Damit mussten pro Frame nur noch 20 statt 64 Bit eingebettet werden. Anschließend wurde der markierte Vektor ausgelesen.

Es wurde zunächst die Robustheit gegenüber inhalts-erhaltenden Maßnahmen, also verlustbehaftete Kompression und Skalierung, getestet. Aus Abbildung 7.13 wird ersichtlich, dass das Verfahren über eine sehr gute Ausleserate verfügt. Sie schwankt zwischen 69,72% bei der Kompression des Videos „Fußball 1“ und 100%.

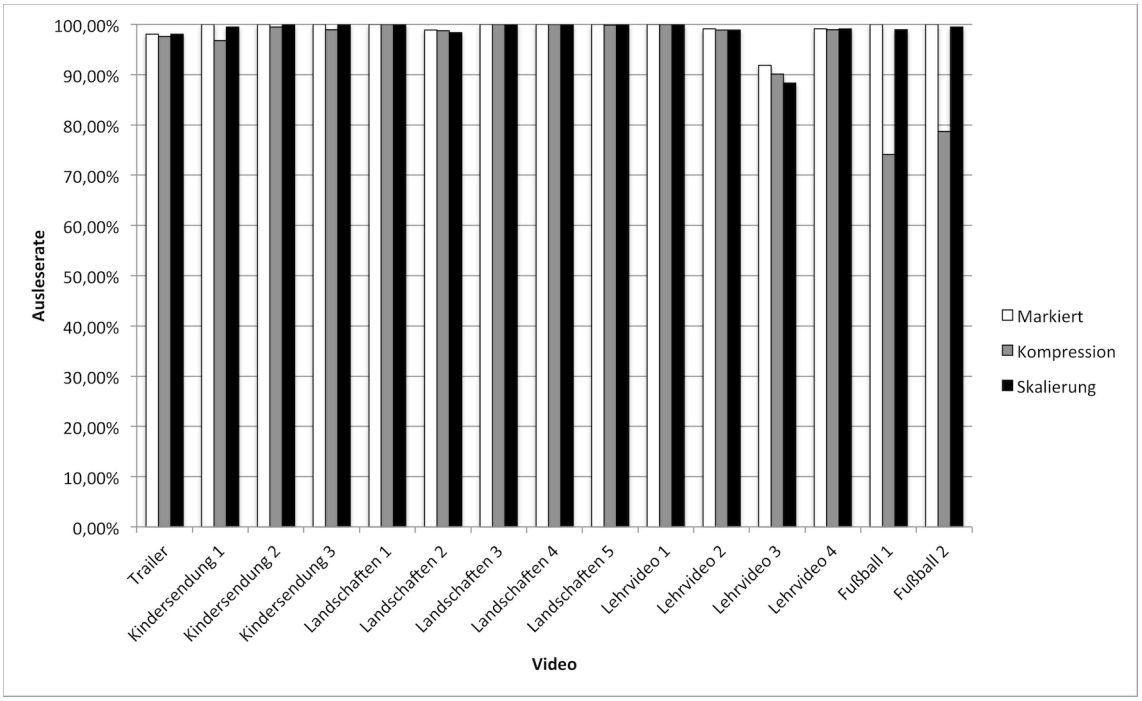


Abbildung 7.13: Ausleserate nach Markierung und inhalts-erhaltenden Maßnahmen

In einer zweiten Testreihe wurde die Robustheit gegenüber inhalts-verändernden Maßnahmen untersucht. Die Ergebnisse sind aus Abbildung 7.14 zu entnehmen. Auch hier ist ersichtlich, dass selbst bei Veränderungen, die 20% des Frame-Inhalts betreffen, also dem Austausch zweier Blöcke mit einer jeweiligen Größe von 10%, das Wasserzeichen immer noch sehr gut ausgelesen werden kann. Die Ausleserate schwankt zwischen 89,60% beim Einfügen von Blöcken in „Lehrvideo 3“ und 100%.

Es wird auch ersichtlich, dass durch die inhalts-verändernden Maßnahmen, die sich hier über die gesamten Videos auf verschiedene Bereiche erstreckten, das eingebettete Wasserzeichen nur mininmal beeinflusst wurde. Die Beschädigung der eingebetteten Merkmalsvektoren, die wir in Kapitel 6 in den jeweiligen Abschnitten erwartet hatten, ist also nicht eingetreten.

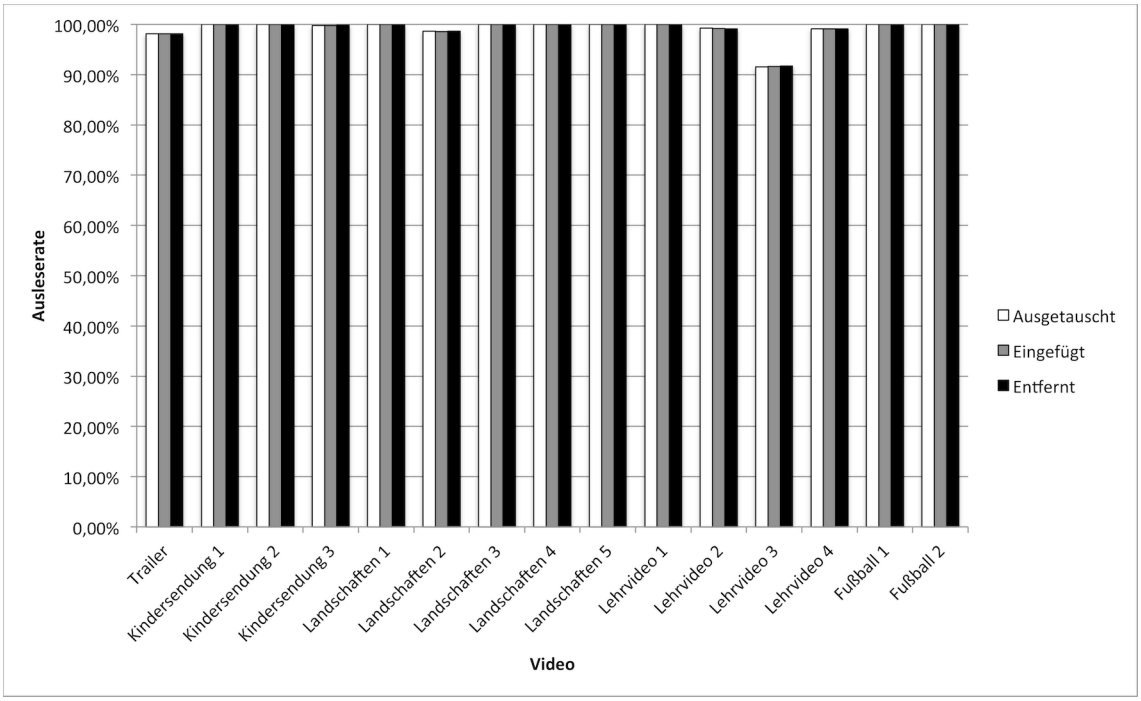


Abbildung 7.14: Ausleserate nach inhalts-verändernden Maßnahmen

7.3 Analyse des Gesamtkonzepts

In der letzten Testreihe untersuchten wir, wie das robuste Wasserzeichenverfahren und das Verfahren zur Generierung von Merkmalsvektoren zusammenarbeiten. Dazu wurden nach der Manipulation die eingebetteten Vektoren mit den aktuellen Merkmalsvektoren verglichen.

Zunächst wurden wieder inhalts-erhaltende Manipulationen untersucht, deren Ergebnisse in Abbildung 7.15 dargestellt sind. Neben den in Kapitel 6.1.3 vorgestellten Manipulationen wurde zusätzlich die Total Rejection Rate (TRR) nach einer Markierung untersucht. Eine Markierung stellt einen zustätzlichen Formatkonvertierungsprozess dar. Daher sind die Robustheitsanforderungen noch einmal für das inhalts-fragile Merkmal gestiegen. Aufgrund dieser Erkenntnis und weiterer Tests wurde ein neuer Parametersatz verwendet, der diesen Anforderungen besser gewachsen ist. Wir verwendeten eine Blockgröße mit einer Auflösung von 16×16 Pixeln, einer Gruppengröße von 2 Blöcken pro Gruppe und einem Quantisierungsfaktor $QF = 1$. Der zeitliche Filter hatte eine Länge von 2 Sekunden. Aus den Ergebnissen in Abbildung 7.15 ziehen wir folgende Erkenntnisse:

- Die TRR schwankt stark in Abhängigkeit des Videos und der Manipulation. Auffällig sind die starken Ausreißer bei der Kompression auf 1.500 kBit/s. Dies betrifft aber nur die Fußball-Szenen. Hier sind weitere Untersuchungen notwendig um herauszufinden, warum gerade hier die TRR so ansteigt.
- Beim Video „Lehrvideo 3“ ist durchschnittlich die höchste TRR festgestellt worden. Auch hier muss untersucht werden, welche Eigenschaften des Videos dafür verantwortlich waren.

Die Abbildungen 7.16, 7.18 und 7.20 zeigen die Correct Rejection Rate für das Austauschen, Einfügen und Entfernen von Blöcken. Die Ergebnisse schwanken zwischen durchschnittlich 43,37% (Entfernen von Blöcken der Größe 5%) und 74,15% (Austausch von Blöcken der Größe 10%). Wieder wirkte sich eine Manipulation von größeren Flächen positiv auf die CRR aus.

Die Abbildungen 7.17, 7.19 und 7.21 zeigen die False Rejection Rate für die inhaltsverändernden Maßnahmen. Daraus ist abzulesen, dass die FRR unabhängig von der Manipulation ist. Sie schwankt zwischen durchschnittlich 5,29% (Entfernen von Blöcken der Größe 10%) und 6,00% (Austausch von Blöcken der Größe 10%). Hier fällt auf, dass die höchsten Fehlerraten bei den Videos „Landschaften 5“ und „Lehrvideo 3“ auftraten. Auch hier bleibt wieder als spätere Forschungsaufgabe zu untersuchen, welche Eigenschaften der Videos für diese Fehlerraten verantwortlich waren.

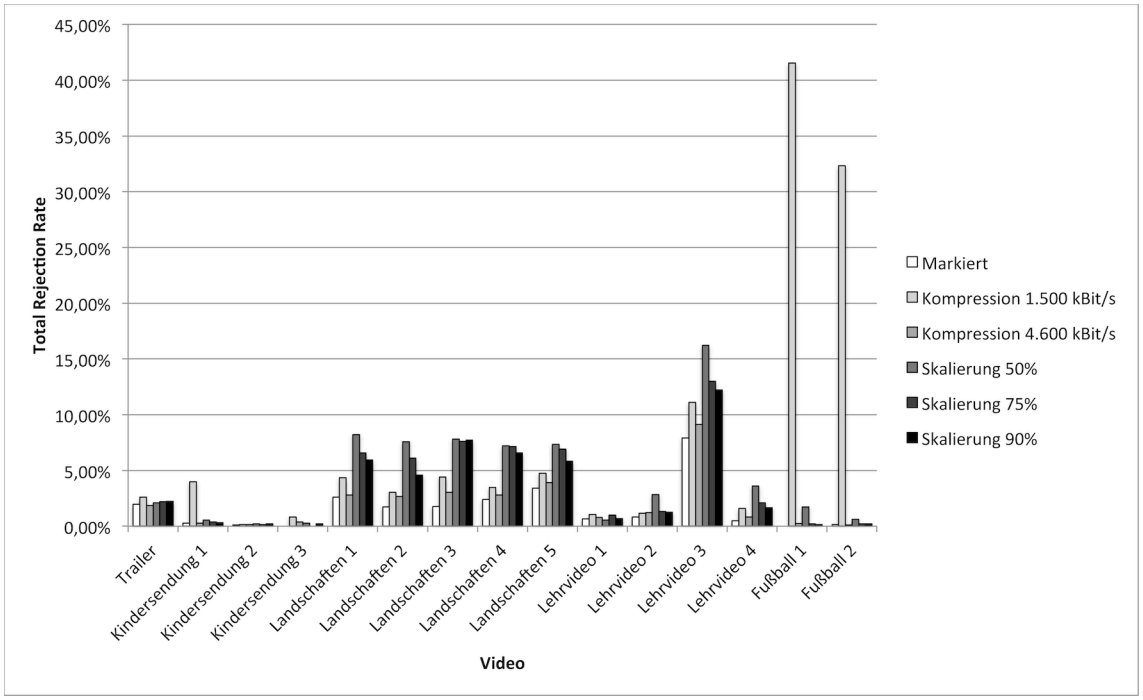


Abbildung 7.15: TRR des Verfahrens

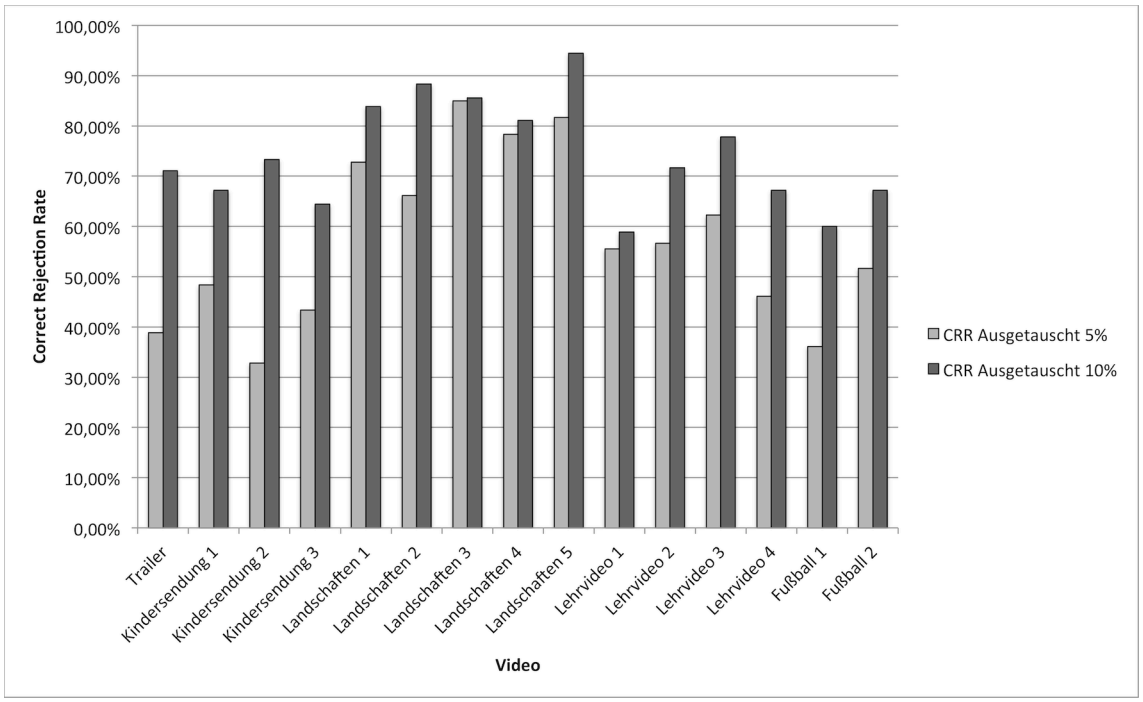


Abbildung 7.16: CRR des Verfahrens gegenüber dem Austausch von Blöcken

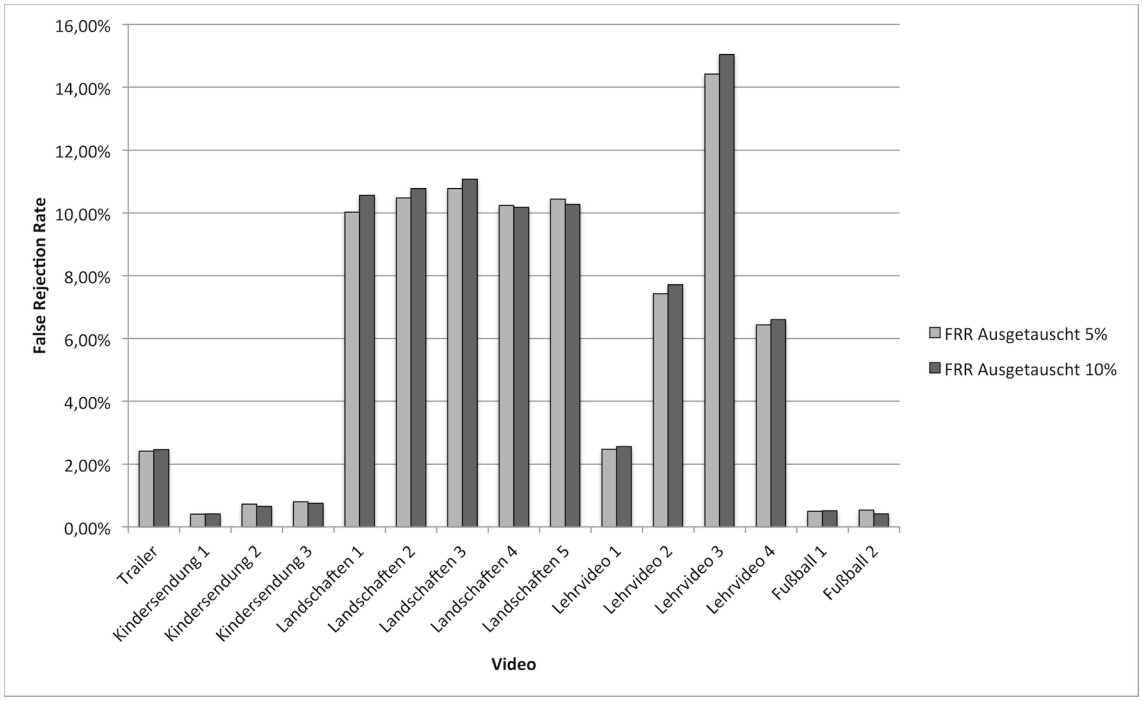


Abbildung 7.17: FRR des Verfahrens gegenüber dem Austausch von Blöcken

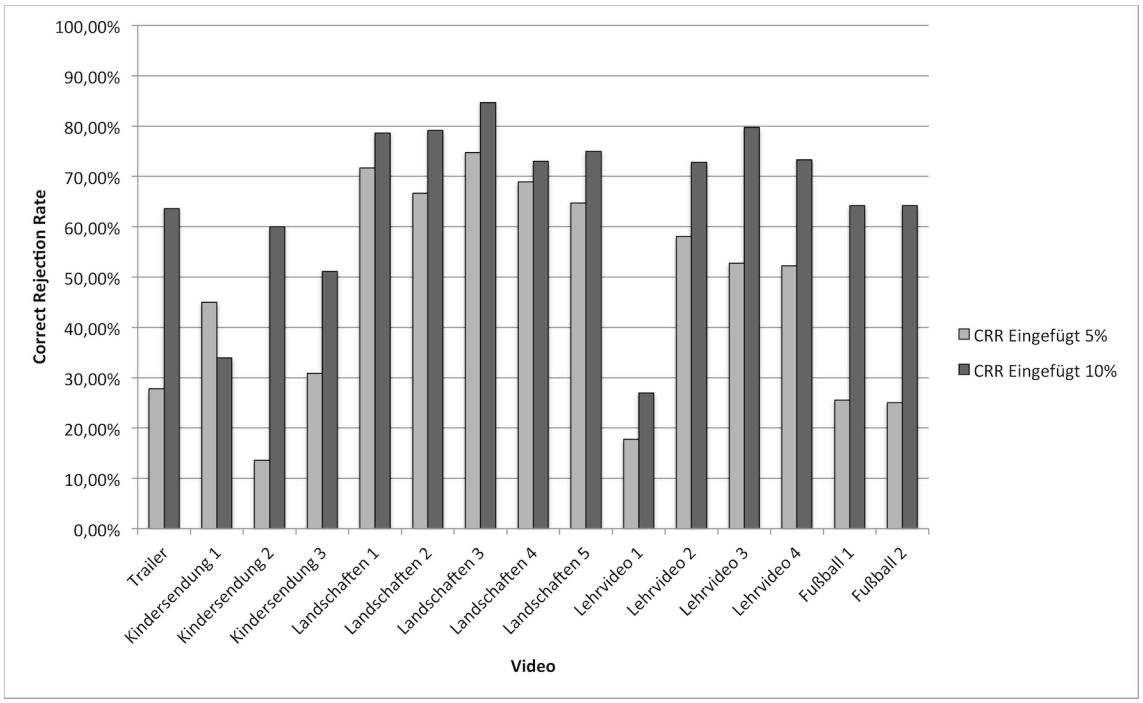


Abbildung 7.18: CRR des Verfahrens gegenüber dem Einfügen von Blöcken

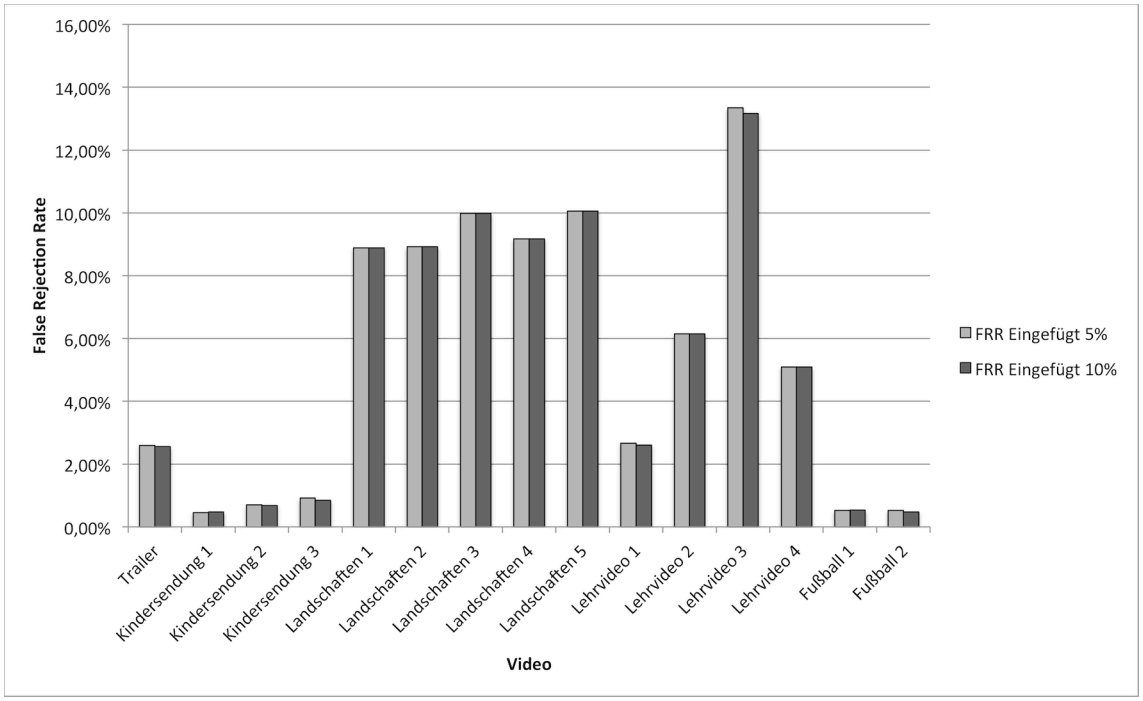


Abbildung 7.19: FRR des Verfahrens gegenüber dem Einfügen von Blöcken

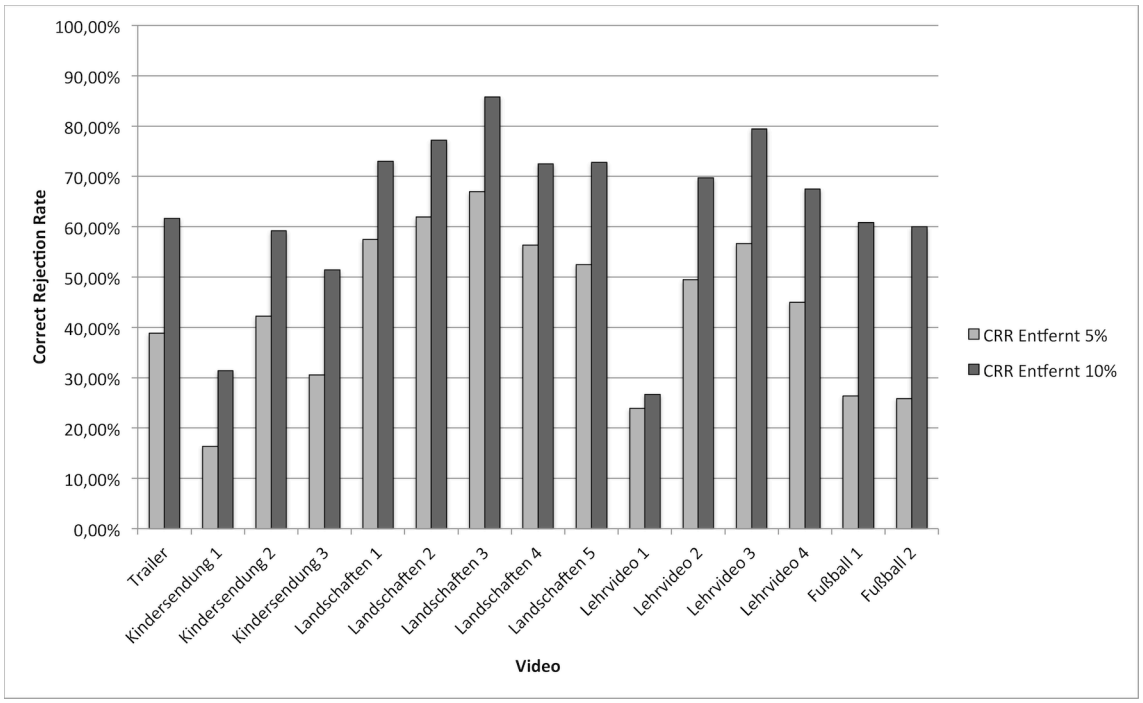


Abbildung 7.20: CRR des Verfahrens gegenüber dem Entfernen von Blöcken

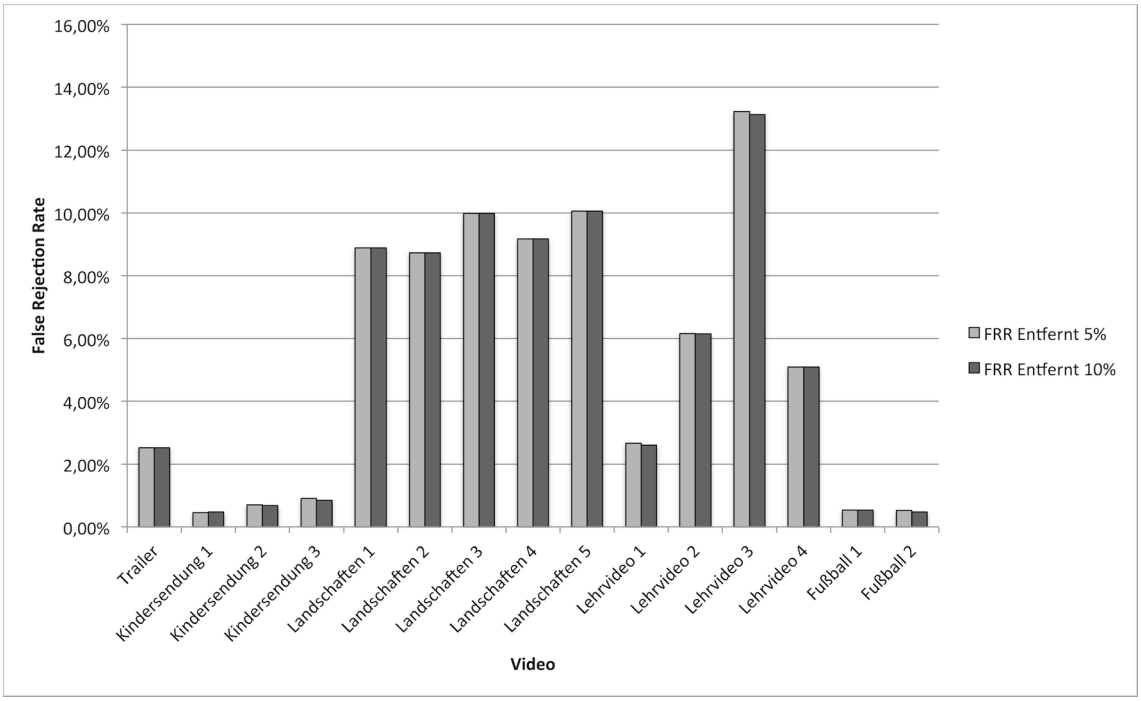


Abbildung 7.21: FRR des Verfahrens gegenüber dem Entfernen von Blöcken

7.4 Zusammenfassung

In diesem Kapitel haben wir gezeigt, dass der Konzeptentwurf aus Kapitel 4 umsetzbar ist. Die Merkmalsvektoren mit einer Länge von 64 Bit konnten robust als Wasserzeichen eingebettet werden. Das robuste Wasserzeichenverfahren zeigte dabei eine sehr gute Robustheit sowohl gegenüber den inhalts-erhaltenden als auch gegenüber den inhalts-verändernden Maßnahmen. Hier hat es unsere ursprünglichen Erwartungen hinsichtlich Kapazität und Robustheit übertroffen. Der Grund ist in der Teilung der Nachricht zu finden. Dadurch musste nicht der gesamte Merkmalsvektor in jedes Frame eingebettet werden und die notwendige Kapazität pro Frame sank erheblich.

Die Sensitivität des inhalts-beschreibenden Merkmals erwies sich als gut. Auffällig ist die gesunkene Robustheit des Entropieverfahrens durch die Formatumwandlungen, die mit der Markierung einhergeht. Hier ist eine Nachbesserung der Parameterauswahl erforderlich um diesem Umstand Rechnung zu tragen und die Robustheit zu verbessern. Die Verbesserung der Sensitivität und die Robustheit ist ein zukünftiger Forschungsgegenstand.

Kapitel 8

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde die Notwendigkeit des Integritätsschutzes von Videodaten diskutiert. Es wurden verschiedene Szenarien definiert, in denen die Unversehrtheit der Daten benötigt wird, z.B. im Rahmen von Videoüberwachung. Es wurde ein Überblick über verschiedene Ansätze zum Integritätsschutz gegeben. Anschließend haben wir ein Konzept entworfen, welches mit Hilfe eines robusten Wasserzeichenverfahrens inhalts-beschreibende Merkmale in das Video einbettet.

Als Basis für das Konzept wurde ein robustes Wasserzeichen von Fridrich und Dittmann weiterentwickelt. Verbesserungen wurden im Bereich der Robustheit und der Kapazität erreicht. Das Verfahren verfügt nun über eine Kapazität von 64 Bit/s und zeigt gute Robustheit gegenüber verlustbehafteter Kompression mit Formatumwandlung und Skalierung. Darüber hinaus wurde ein weiteres Verfahren entwickelt, das ebenfalls über eine gute Robustheit und Kapazität verfügt.

Es wurden vier Verfahren zur Generierung von Merkmalsvektoren entwickelt. Zwei dieser Verfahren bilden nicht-inhaltliche Abhängigkeiten auf Merkmalsvektoren ab (Energiedifferenz, Entropie), während die beiden anderen Verfahren auf inhaltlichen Abhängigkeiten durch Interest-Operatoren basierten (Moravec, Scale Invariant Feature Transform). Wir konnten zeigen, dass sowohl inhaltliche als auch nicht-inhaltliche Abhängigkeiten zur Verwendung geeignet sind. Die Verfahren verfügen über eine unterschiedliche Komplexität, so dass die Berechnungszeit variiert. Die Länge der Merkmalsvektoren ist in Abhängigkeit der Kapazität des zugrundeliegenden Wasserzeichenverfahrens ebenfalls parametrisierbar.

Die Verfahren wurden verschiedenen Tests hinsichtlich ihrer Robustheit gegenüber inhalts-erhaltenden und ihrer Sensitivität hinsichtlich inhalts-verändernder Maßnahmen unterzogen. Dazu wurde ein Video mit verschiedenen Charakteristiken verlustbehafteter Kompression, Skalierung, Formatumwandlung und verschiedenen Blockmanipulationen unterzogen. Aus den modifizierten Videos wurden Merkmalsvektoren generiert und mit den Originalvektoren verglichen. Daraus wurden die Total Rejection Rate, die Correct Rejection Rate und die False Rejection Rate berechnet. Diese Tests ermöglichten eine Vergleichbarkeit der Verfahren. Es wurde ersichtlich,

dass das Entropieverfahren über die beste Kombination aus Robustheit und Sensitivität verfügte. Daher wurde es für die finale Implementierung als Grundlage verwendet.

In der finalen Evaluierung wurden 15 weitere Videos mit dem Integritätswasserzeichen markiert und dann wiederum inhalts-erhaltenden und inhalts-verändernden Maßnahmen unterzogen. Die Testergebnisse bestätigten die gute Robustheit des zugrundeliegenden Wasserzeichenverfahrens. Ebenso wurde die gute Sensitivität des Entropieverfahrens bestätigt. Auffällig war die stark gesunkene Robustheit des Verfahrens gegenüber mehrfacher Formatumwandlung. Dies ist Gegenstand zukünftiger Forschungsarbeiten.

Für weitere Forschungen auf dem Gebiet der Integritätswasserzeichen für Videodaten empfehlen wir eine Konzentration auf die Rotations-, Skalierungs- und Verschiebungsinvarianz sowohl der Merkmalsvektoren als auch der robusten Wasserzeichen. Aufgrund der Blockbasiertheit beider Elemente unseres Konzepts kann es schnell zu einer Desynchronisation kommen. Darüber hinaus sollte die Länge der Merkmalsvektoren reduziert werden können und die Sensitivität des Verfahrens gerade bei kleineren Modifikationen erhöht werden. Es ist darüber hinaus ersichtlich, dass die Verfahren größeren Tests unterzogen werden müssen, um die Übertragbarkeit auf Videos mit anderen Charakteristiken zu gewährleisten.

Literaturverzeichnis

[Buc10] Johannes Buchmann. *Einführung in die Kryptographie*. Springer, Berlin, 2010.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8 (6), 1986.

[CMB+07] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2007.

[CSST02] Mehmet U. Celik, Eli S. Saber, Gaurav Sharma, and A. Murat Tekalp. Video authentication with self-recovery. In E. J. Delp III and P. W. Wong, editors, *Electronic Imaging: Security and Watermarking of Multimedia Contents IV*, volume 4675 of *Proceedings of SPIE*, pages 531–541, 2002.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley & Sons, 1991.

[CW01] B. Chen and G.W. Wornell. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47 (4):1423 – 1443, 2001.

[Dit00] Jana Dittmann. *Digitale Wasserzeichen*. Springer, 2000.

[DS00] J. Dittmann and M. Steinebach. Manipulationserkennung bei digitalem Bildmaterial mit fragilen Wasserzeichen. *Datenschutz und Datensicherheit*, 24 (10), 2000.

[DSR+00] J. Dittmann, M. Steinebach, I. Rimac, S. Fischer, and R. Steinmetz. Combined video and audio watermarking: Embedding content information in multimedia data. In E. J. Delp III and P. W. Wong, editors, *Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents II*, volume 3971 of *Proceedings of SPIE*, pages 455–464, 2000.

[DSS98] J. Dittmann, M. Stabenau, and R. Steinmetz. Robust MPEG video watermarking technologies. In *Proceedings of the sixth ACM international conference on Multimedia*, 1998.

[DSS99] J. Dittmann, A. Steinmetz, and R. Steinmetz. Content-based digital signature for motion pictures authentication and content-fragile watermarking. In A. Steinmetz, editor, *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, volume 2, pages 209–213, 1999.

[DTS04] Yuewei Dai, Stefan Thiemert, and Martin Steinebach. Feature-based watermarking scheme for MPEG-I/II video authentication. In Edward J. Delp III and Ping Wah Wong, editors, *Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents VI*, 2004.

[EG01] J.J. Eggers and B. Girod. Blind watermarking applied to image authentication. In B. Girod, editor, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1977–1980, 2001.

[ESC+04] Ö. Ekici, B. Sankur, B. Coşkun, U. Naci, and M. Akcay. Comparative evaluation of semifragile watermarking algorithms. *Journal of Electronic Imaging*, 13(1):209, 2004.

[FG87] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proceedings of ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, 1987.

[FG00] J. Fridrich and M. Goljan. Robust hash functions for digital watermarking. In *Proceedings of the International Conference on Information Technology: Coding and Computing*, pages 178–183, 2000.

[FGD02] J. Fridrich, M. Goljan, and R. Du. Cryptanalysis of the Yeung-Mintzer fragile watermarking technique. *Electronic Imaging*, 11:262–274, 2002.

[FGM00] J. Fridrich, M. Goljan, and N. Memon. Further attacks on Yeung-Mintzer fragile watermarking scheme. In E. J. Delp III and P. W. Wong, editors, *Electronic Imaging: Security and Watermarking of Multimedia Contents II*, volume 3971 of *Proceedings of SPIE*, pages 428–437, 2000.

[Fri93] Gary L Friedman. The trustworthy digital camera: restoring credibility to the photographic image. *IEEE Transactions on Consumer Electronics*, 39(4):905–910, 1993.

[Fri97] J. Fridrich. Technical Report: Methods for data hiding, 1997.

[Fri02] J. Fridrich. Security of fragile authentication watermarks with localization. In E. J. Delp III and P. W. Wong, editors, *Electronic Imaging: Security and Watermarking of Multimedia Contents IV*, volume 4675 of *Proceedings of SPIE*, pages 691–700, 2002.

[FTM98] A. Ferman, A. Tekalp, and R. Mehrota. Effective content representation for video. In *Proceedings of the International Conference on Image Processing, ICIP*, volume 3, pages 521 – 525, 1998.

[Gol82] S. W. Golomb. *Shift Register Sequences.* Aegean Park Press, 1982.

[HHP08] Weilin Huang, Anthony Ho, and Vinod Pankajakshan. Watermarking-Based Content Authentication of Motion-JPEG Sequences. In *Proceedings of the 5th International Conference on Visual Information Engineering, VIE*, pages 813 – 818, 2008.

[HM00] M. Holliman and N. Memon. Counterfeiting Attacks on Oblivious Block-Wise Independent Invisible Watermarking Schemes. *IEEE Transactions on Image Processing*, 9 (3)(s):432–441, 2000.

[KH99] D. Kundur and D. Hatzinakos. Digital watermarking for telltale tamper proofing and authentication. In *Proceedings of the IEEE*, volume 87 (7), pages 1167 – 1180, 1999.

[KR00] N. Kaewamnerd and K. R. Rao. Wavelet based image adaptive watermarking scheme. *Electronics Letters*, 36(4):312–313, 17 Feb. 2000.

[KTB07] V. Kitanovski, D. Taskovski, and S. Bogdanova. Semi-Fragile Watermarking Scheme for Authentication of MPEG-1/2 Coded Videos. In *Proceedings of the 14th International Workshop on Systems, Signals and Image Processing, 2007 and the 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services*, pages 225 – 228, 2007.

[LC98] C.-Y. Lin and S.-F. Chang. A Robust Image Authentication Method Surviving JPEG Lossy Compression. In *Storage and Retrieval of Image/Video Databases*, Proceedings of SPIE, pages 296–307, 1998.

[LC99] C.-Y. Lin and S.-F. Chang. Issues and solutions for authenticating MPEG video. In E. J. Delp III and P. W. Wong, editors, *Electronic Imaging: Security and Watermarking of Multimedia Contents*, volume 3657 of *Proceedings of SPIE*, pages 54–65, 1999.

[LC00] C.-Y. Lin and S.-F. Chang. Semi-Fragile Watermarking for Authenticating JPEG Visual Content. In E. J. Delp III and P. W. Wong, editors, *Electronic Imaging: Security and Watermarking of Multimedia Contents II*, volume 5306 of *Proceedings of SPIE*, pages 325–335, 2000.

[LC01] Ching-Yung Lin and Shih-Fu Chang. A robust image authentication method distinguishing JPEG compression from malicious manipulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(2):153–168, 2001.

[LLB99] G. C. Langelaar, R. L. Lagendijk, and J. Biemond. Watermarking by DCT coefficient removal : A statistical approach to optimal parameter settings. In E. J. Delp III and P. W. Wong, editors, *Electronic Imaging: Security and Watermarking of Multimedia Contents*, volume 3657 of *Proceedings of SPIE*, pages 2–13, 1999.

[LMT00] T. H. Lan, M.F. Mansour, and A. H. Tewfik. Robust high capacity data embedding. In *Proceedings of the International Conference on Image Processing*, volume 1, pages 581 – 584, 2000.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LPD00] E. T. Lin, C. Podilchuk, and E. J. Delp. Detection of Image Alterations Using Semi-Fragile Watermarks. In E. J. Delp III and P. W. Wong, editors, *Electronic Imaging: Security and Watermarking of Multimedia Contents II*, volume 3971 of *Proceedings of SPIE*, pages 152–163, 2000.

[ME01] B.G. Mobasseri and A. T. Evans. Content-dependent video authentication by self-watermarking in color space. In E. J. Delp III and P. W. Wong, editors, *Electronic Imaging: Security and Watermarking of Multimedia Contents III*, volume 4314 of *Proceedings of SPIE*, pages 35–44, 2001.

[Mor77] Hans P. Moravec. Towards Automatic Visual Obstacle Avoidance. In *Proceedings of 5th Int. Joint Conference on Artificial Intelligence*, volume 2, 1977.

[MSCS06] Kurato Maeno, Qibin Sun, Shih-Fu Chang, and Masayuki Suto. New semi-fragile image authentication watermarking techniques using random bias and nonuniform quantization. *IEEE Transactions on Multimedia*, 8 (1):32 – 45, 2006.

[MSS00] B.G. Mobasseri, M.J. Sieffert, and R.J. Simard. Content authentication and tamper detection in digital video. In M.J. Sieffert, editor, *Proceedings of the International Conference on Image Processing*, volume 1, pages 458–461, 2000.

[OKH01] Job Oostveen, Ton Kalker, and Jaap Haitsma. Visual hashing of digital video: applications and techniques. In *Proceedings of SPIE: Applications of digital image processing XXIV*, volume 4472, pages 121–131, 2001.

[OP98] Joseph J. K. O'Ruanaidh and Thierry Pun. Rotation, scale and translation invariant spread spectrum digital image watermarking. *Signal Processing*, 66(3):303 – 317, 1998.

[PF04] A. C. Popescu and H. Farid. Exposing Digital Forgeries by Detecting Duplicated Image Regions. In *Dartmouth Computer Science, Technical Report TR2005-515*, 2004.

[Pro08] *Protector - Die europäische Fachzeitschrift für Sicherheit.* I.G.T. Informationsgesellschaft Technik mbH, München, März 2008.

[PRSM05] D. Pröfrock, H. Richter, M. Schlauweg, and E Müller. H.264/AVC video authentication using skipped macroblocks for an erasable watermark. In *Proceedings of Conference Visual Communications and Image Processing*, volume 5960, 2005.

[Que01] M. P. Queluz. Authentication of digital images and video: Generic models and a new contribution. *Signal Processing: Image Communication*, 16 (5):461–475, 2001.

[Sch96] Bruce Schneier. *Angewandte Kryptographie . Protokolle, Algorithmen und Sourcecode in C*. Addison-Wesley, 1996.

[Sch99] Oliver Schimmel. *Dreidimensionale Szenenmodellierung durch monokulare Exploration mit einem mobilen Roboter*. Universität Tübingen, Institut für Informatik, 1999.

[Sha48] Claude Elwood Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[Ste04] Martin Steinebach. *Digitale Wasserzeichen für Audiodaten*. Shaker Verlag, Aachen, 2004.

[Sti04] Stiftung Haus der Geschichte der Bundesrepublik Deutschland, editor. *Bilder, die lügen: X für U*. Bouvier, 2004.

[TS10] Stefan Thiemert and Martin Steinebach. SIFT features in semi-fragile video watermarks. In Nasir D. Memon, Jana Dittmann, Adnan M. Alattar, and Edward J. Delp III, editors, *Electronic Imaging: Media Forensics and Security II*, volume 7541, page 75410, 2010.

[TSL09] Stefan Thiemert, Martin Steinebach, and Huajian Liu. Digital watermarking for digital cinema. In Edward J. Delp III, Jana Dittmann, Nasir D. Memon, and Ping Wah Wong, editors, *Electronic Imaging: Media Forensics and Security XI*, volume 7254 of *Proceedings of SPIE*, page 72540V, 2009.

[TSS05] Stefan Thiemert, Hichem Sahbi, and Martin Steinebach. Applying interest operators in semi-fragile video watermarking. In Edward J. Delp III and Ping Wah Wong, editors, *Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681 of *Proceedings of SPIE*, pages 353–362, 2005.

[TSS06] Stefan Thiemert, Hichem Sahbi, and Martin Steinebach. Using entropy for image and video authentication watermarks. In Edward J. Delp III and Ping Wah Wong, editors, *Electronic Imaging: Security, Steganography, and Watermarking of Multimedia Contents VIII*, Proceedings of SPIE, pages 470–479, 2006.

[TTS08] Stefan Thiemert, Daniel Thiemert, and Martin Steinebach. Sicherung der Echtheit von Videodaten mit digitalen Wasserzeichen. In Patrick Horster, editor, *D.A.CH. Security 2008 - Bestandsaufnahme, Konzepte, Anwendungen*, 2008.

[TVDS04] Stefan Thiemert, Thomas Vogel, Jana Dittmann, and Martin Steinebach. A High-Capacity Block Based Video Watermark. In *Proceedings of the 30th EUROMICRO-Conference*, 2004.

[WF06] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double MPEG compression. In *Proceedings of the ACM Multimedia and Security Workshop*, 2006.

[Won98] P. W. Wong. A Watermark for Image Integrity and Ownership Verification. In *Proceedings of IS&T's Image Processing, Image Quality, Image Capture, Systems Conference*, pages 374–379, 1998.

[YLL01] G.-J. Yu, C.-S. Lu, and H.-Y. M. Liao. Mean Quantization-based Fragile Watermarking for Image Authentication. *Opt. Eng*, 40 (7):1396 – 1408, 2001.

[YM97] M. M. Yeung and F. Mintzer. An invisible watermarking technique for image verification. In *Proceedings of the International Conference on Image Processing*, volume 2, pages 680–683, 1997.

[YY02] P. Yin and H. H. Yu. A semi-fragile watermarking system for MPEG video authentication. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, volume 4, 2002.

[YY08] Zhi Yuan and Sheng Yan, Peimin Li. Super resolution based on scale invariant feature transform. In *Proceedings of the International Conference on Audio, Language and Image Processing*, pages 1550–1554, 2008.

[ZK95] J. Zhao and E. Koch. Embedding Robust Labels into Images for Copyright Protection. In *Proceedings of the International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, 1995.